Real-Time Simulation of Flexible Multibody Systems in Vehicle Dynamics

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von
Alexander Schmitt
aus Emmendingen

2019

1. Gutachter: Prof. Dr.-Ing. Robert Seifried
2. Gutachter: Jun.-Prof. Dr.-Ing. Jörg Fehr

Tag der mündlichen Prüfung: 25. 10. 2019

MuM Notes in Mechanics and Dynamics

Editor: Prof. Dr.-Ing. Robert Seifried
Hamburg University of Technology
Institute of Mechanics and Ocean Engineering (MuM)
www.tuhh.de/mum

Volume 3
Alexander Schmitt
“Real-Time Simulation of Flexible Multibody Systems in Vehicle Dynamics”
Hamburg, 2019

DOI: 10.15480/882.2510

Preface

This work results from the research I conducted during my occupation as a research associate at the Institute of Mechanics and Ocean Engineering (MuM) at the Hamburg University of Technology.
I want to thank my advisor, Prof. Dr.-Ing. Robert Seifried for giving me the opportunity to return back to science at the University of Siegen and the chance to continue my plans at the Hamburg University of Technology. Without his continuous support I would not have been able to create this work. Furthermore I want to thank Jun-Prof. Dr.-Ing. Jörg Fehr for his attribution as second advisor of this work and Prof. Dr.-Ing. Thomas Rung for supervising the doctoral examination.
Significant parts of this work could be achieved during my research visit at the University of California at Berkeley. I want to thank Prof. Oliver O'Reilly for giving me the opportunity to visit the dynamics lab at UC Berkeley and conduct my research on quaternions at this inspiring place.
Sincere thanks goes to the whole team of the Institute of Mechanics and Ocean Engineering for the good collaboration during my time in Hamburg, for helping me whenever help was needed and also for the countless coffee break conversations.
Last but not least I want to thank my parents Ursula and Konrad Schmitt for the constant encouragement during my research in Hamburg and my sister Dr. Katrin Schmitt for telling me how things work in science.

To my family.

Contents

Introduction

When looking at a car, almost everybody would assume that the vehicle body can be treated as a rigid body. The times are gone for good when convertible roofs did not close if the vehicle was parked on a curb or a drive on cobblestone road caused deafening noise inside the vehicle. However, although modern cars show very low chassis flexibility, lightweight design, increasing comfort demands and complex body shapes require the investigation of structural flexibilities. New challenges like drivetrain electrification or energy efficiency will add even more complex requirements to the design process.
Structural flexibilities in vehicle bodies are influencing various fields of vehicle development. Depending on the loads on the vehicle several effects can occur. Its chassis may deform statically, vibrations uncomfortable for passengers or the environment may be emitted or even permanent damage to vehicle components may be caused. The vehicle design process is influenced greatly by the effects of body flexibilities to ensure a safe and comfortable vehicle.
In modern vehicle development real-time (RT) vehicle models are used in various stages of the process. Electronic control units are nowadays an integral part of every vehicle and will even gain control of the vehicle itself in autonomous driving scenarios. This requires an enormous amount of hardware and software to be tested under realistic conditions. Hardware-in-the-Loop (HiL) systems are used for component tests and require RT capable vehicle models to generate realistic communication bus and sensor data for the tested components. Control algorithms increasingly use virtual models of the system to be controlled, requiring RT capable vehicle models as well. Even though not really a scientific field of application, modern racing simulations use astonishingly detailed vehicle models to create a realistic feeling of virtual vehicles in real-time.
The goal of this work is to merge the vast fields of RT capable simulation methods and structural flexibilities of vehicle components. The application to a vehicle body can be seen as an exemplary implementation. The developed methods can be used in other fields like aviation or marine systems as well.

1.1 Focus of This Work

This work focuses on the modeling and efficient simulation of structural flexibilities within multibody systems with the application of the developed methods on a full vehicle model. The topic of RT simulation is already well established

for rigid multibody systems [2, 16] and the application to vehicle dynamics simulations [14]. However, structural flexibilities are either neglected in current approaches or simple representations like the lumped-mass approach are used for various components like tires [40] or suspension bushings [47].
Generating a realistic full vehicle model will result in a detailed representation of suspension kinematics. Most modern cars feature an independent suspension setup that is attached to the chassis with a complex constraint situation. Modeling a road vehicle results in a multibody system (MBS) which can be modeled preferably with generalized coordinates or cartesian coordinates. When considering only rigid bodies, most constraints and resulting kinematic loops can be resolved analytically to a minimal set of independent coordinates [44, 56]. However, the occurring kinematic loops are not resolvable analytically when considering structural flexibilities, hence a flexible multibody system in general results in a system of differential-algebraic equations (DAEs). While the solution of DAEs can be carried out easily with commercial toolboxes, simulation of DAEs in RT is very sophisticated and requires special solvers [1, 37].
The information on deformation behavior of vehicle components is either governed by vehicle tests or from simulation methods, mostly by continuum-based approaches. In order to use realistic vehicle data, finite element (FE) models can be used to obtain information on static and modal deformation [35]. Since FE models usually show a very large number of degrees of freedom, they cannot be used for RT purposes. Model order reduction methods are a well-established method to extract the information on relevant deformation modes from continuum based models [18, 32].
The outcome of this work is a RT capable vehicle model with structural flexibility, using efficient simulation methods to implement differential-algebraic equations in real-time (RT). FE data of a state-of-the-art vehicle is used together with model order reduction methods to implement structural flexibility in the vehicle model.

1.2 Structure

Two main topics characterize the content of this work, efficient and RT capable formulations of flexible multibody systems and flexible body definition with quaternions. The unique combination of employing quaternions to constrained flexible multibody systems allows the RT capable integration even of complex and highly constrained multibody systems.
Real-time simulation requires the solution of DAEs with an a priori known simulation time. In chapter 2 the basics of constrained multibody systems are outlined with a focus on rotation formulations with quaternions. Quaternions are a group of complex numbers in 4D that can be used to describe rotations in space. Their application in constrained mechanical systems allows the analytical sensitivity analysis for constraint equations and applied forces. Common constraints that are used in vehicle technology are stated for flexible MBS and a sensitivity

analysis is performed to govern the equations on velocity and acceleration level. This allows time integration with explicit or semi-implicit solvers and fixed step size, which are introduced as well. RT capable constraint stabilization methods are shown to ensure compliance of constraint equations on position level. Flexible MBS with small, elastic deformations are usually formulated with the floating frame of reference (FFoR) approach [57, 59]. The FFoR approach can cause numerical problems for integration with large step sizes. A mixed reference frame approach is used to allow a stable time integration of flexible bodies.
As a by-product to the definition of rotations for flexible multibody system (FMBS) in quaternions the equation of motion (EOM) can be stated solely in terms of quaternion derivatives, which is derived in chapter 3. A small example shows the functionality of this method.
Based on the methods of RT capable FMBS a vehicle model derived from a detailed FE model for crash simulation purposes is set up in chapter 4. The model is combined with a RT capable road and tire model and implemented in a numerical environment that allows deployment of the model on RT platforms in chapter 5. It is validated with a commercial MBS software and several driving maneuvers are performed. The most crucial question of this work - the simulation time required for the flexible MBS - is answered with a performance comparison. Two major simplifications are shown to further improve the efficiency of the model. The Jacobian calculation can be neglected partially, some parts can be kept constant. By introducing generalized coordinates for highly constrained subsystems the system size can be reduced significantly. These improvements help to reduce the computation time significantly, hence enabling the RT capable simulation of complex vehicle models with structural flexibility.
Chapter 6 summarizes the achieved results of this work. It shows options to further improve the methods for RT capable flexible multibody simulation and various fields of application as well.

Real-Time Capable Multibody Simulations

This chapter provides the necessary background for efficient and real-time capable multibody simulation. Since RT simulation methods mostly rely on classical MBS, an introduction to rigid MBS is given first. Thereby a special focus is set on efficient formulation of the necessary equations. Heavy use of quaternions as a rotation formulation is made due to the advantages in constraint definition and sensitivity analysis. The formulations of MBS in generalized coordinates and in redundant Cartesian coordinates with additional constraint equations are presented. For the constraint equations the sensitivity analysis is presented as well.

Real-time capable time integration states high demands on the integration methods. Both for ordinary differential equation (ODE) and DAE a group of numerical solvers is presented that fulfill these demands, besides the modifications that have to be made to the EOM.

FMBS extend rigid MBS with the ability to calculate elastic deformations of some bodies simultaneously to the large, mostly nonlinear, rigid body motion. The FFoR approach is presented as a method to efficiently simulate body flexibilities in real-time, as well as some modifications to improve the efficiency of this method is discussed. The elastic information of FMBS is usually derived from continuum mechanical models. In most cases complex FE models are used and thus a brief introduction to model order reduction methods is given here. This allows the reduction of the model size to a relatively low number of elastic coordinates and hence the implementation in RT capable simulations.

2.1 Basics of Multibody Systems

The dynamics of mechanical systems often are composed of the movement of mostly rigid bodies, ideal constraints and discretized force elements and force fields. The movement may be free in space, resulting in six degrees of freedom (DOF) for a single rigid body. The term multibody system then refers to the constrained movement of several rigid bodies loaded by external, internal and inertial forces, which is mathematically described by the EOM. An exemplary multibody system with typical discrete elements is shown in figure 2.1.

In order to derive the EOM for a comprehensive multibody system, the equations for a single, unconstrained body have to be derived first. Thus, first the

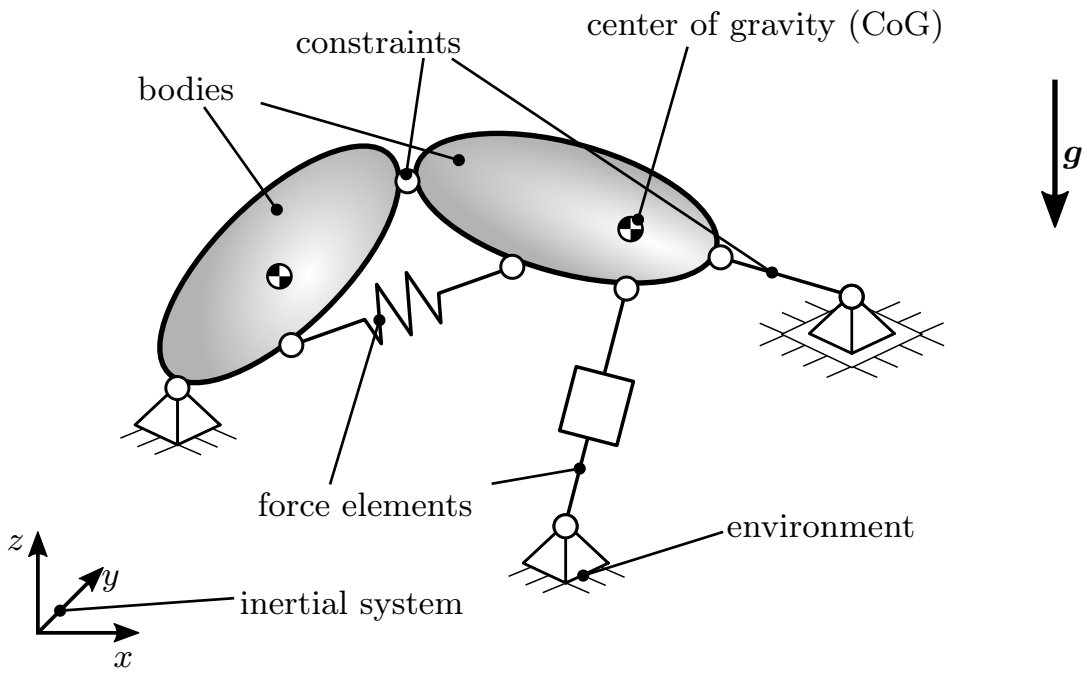


Figure 2.1: Multibody system.

kinematics of mass points and rigid bodies are derived, followed by the kinetic analysis to result in the EOM of the MBS.

2.1.1 Kinematics of a Mass Point

A mass point is the most simple model element moving in space where all mass is concentrated in one point. Its advantage is the simple description since it can only move in translational direction. However, this is a disadvantage as well since for most engineering examples like vehicle motion, the orientation is of great importance. The analysis of rigid and flexible bodies in subsequent sections will show that mass point kinematics are necessary to derive these equations as well. A mass point in space m_i, as shown in figure 2.2 at position $\boldsymbol{\rho}_i$, is loaded by a force $\boldsymbol{F}_i$. This will yield to acceleration of the mass point, hence in a first step a kinematic analysis is necessary.

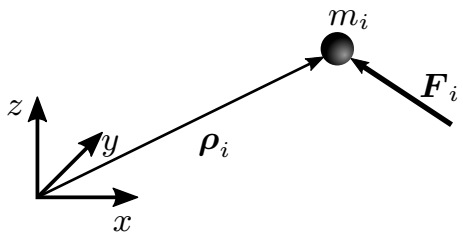


Figure 2.2: Free mass point.

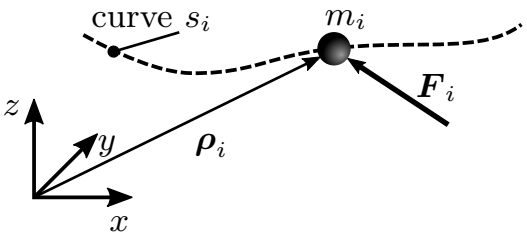


Figure 2.3: Constrained mass point.

For a free mass point, the position vector

$$\boldsymbol{\rho}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{2.1}$$

describes the position in space, each of the vector components denote a DOF. The components of the vector are given in the global reference frame shown in the picture. Here a cartesian reference frame is chosen, other reference frames as polar or cylinder coordinates may be more suitable for problems involving mostly circular motion, e.g. planetary motion. The time derivatives of the position vector then yield

$$\frac{\mathrm{d}\boldsymbol{\rho}_i}{\mathrm{d}t} = \boldsymbol{v}_i = \begin{bmatrix} v_{x,i} & v_{y,i} & v_{z,i} \end{bmatrix}^T = \begin{bmatrix} \frac{\mathrm{d}x_i}{\mathrm{d}t} & \frac{\mathrm{d}y_i}{\mathrm{d}t} & \frac{\mathrm{d}z_i}{\mathrm{d}t} \end{bmatrix}^T \text{ and} \tag{2.2}$$

$$\frac{\mathrm{d}^2\boldsymbol{\rho}_i}{\mathrm{d}t^2} = \boldsymbol{a}_i = \begin{bmatrix} a_{x,i} & a_{y,i} & a_{z,i} \end{bmatrix}^T = \begin{bmatrix} \frac{\mathrm{d}^2x_i}{\mathrm{d}t^2} & \frac{\mathrm{d}^2y_i}{\mathrm{d}t^2} & \frac{\mathrm{d}^2z_i}{\mathrm{d}t^2} \end{bmatrix}^T \tag{2.3}$$

for velocity and acceleration of the mass point.
For the case that the mass point is constrained to a plane or a curve s_i as shown in figure 2.3, the number of DOF of the mass point is reduced. The curve s_i may be used to describe the position coordinates as a function of a coordinate $s\,(t)$ with

$$\boldsymbol{\rho}_i = \begin{bmatrix} x_i(s) & y_i(s) & z_i(s) \end{bmatrix}^T . \tag{2.4}$$

Using the Jacobian

$$\boldsymbol{J}_i = \frac{\partial \boldsymbol{\rho}_i}{\partial s} \tag{2.5}$$

velocity and acceleration can be described as

$$\boldsymbol{v}_i = \boldsymbol{J}_i \dot{s}, \quad \boldsymbol{a}_i = \boldsymbol{J}_i \ddot{s} + \frac{\mathrm{d}\boldsymbol{J}_i}{\mathrm{d}t} s\,. \tag{2.6}$$

This method will be useful when deriving EOM using a minimal number of coordinates, since the method introduced above indicates a explicit description of constraints.

2.1.2 Kinematics of a Rigid Body

A rigid body is an idealized body that neglects all inner deformation, so the distance between two material points of the body is assumed to be constant. This assumption is valid for many mechanical systems and results in rather simple equations for MBS. Is it required to consider deformations within a body, the method of FMBS explained later in this chapter can be utilized. With several material points connected together, the relative position between them becomes significant. Figure 2.4 shows a rigid body with a body origin O′ defined by the vector $\boldsymbol{\rho}_i$ and two points k and l defined relative to the body origin O′ defined by the vectors $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$. A local reference system $K_i = (\mathrm{O}', x', y', z')$ is used to describe the location of points of the body instead of the global system

$K_I = (\mathrm{O}, x, y, z)$. This body can be loaded with forces $\boldsymbol{f}_k$ acting on specific material points and free moments $\boldsymbol{\ell}_i$, which will result in acceleration of the body.

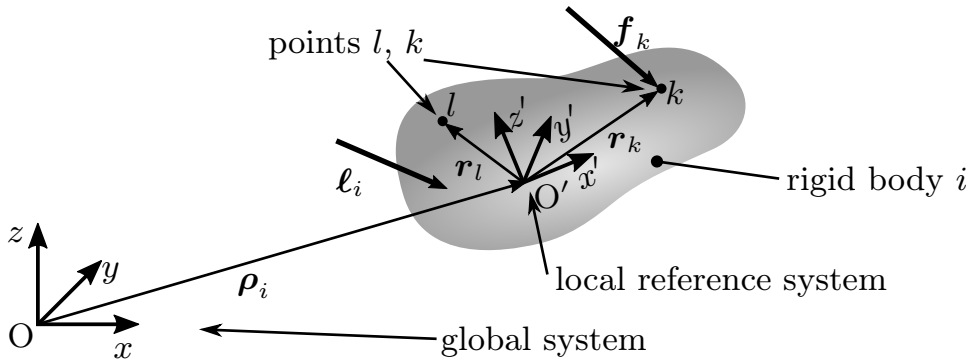


Figure 2.4: Rigid body with local reference system.

The local reference system defined above is not only translated relative to the global inertial system, but also shows a relative orientation or attitude. This introduces three additional DOF for a rigid body, resulting in a total of six DOF for a body moving freely in space.

2.1.3 Orientation Formulations

Vectors defined in the global coordinate system K_I can be transformed into the local reference system of a body i with the coordinate transformation

$$ {}^{i}\boldsymbol{r} = \boldsymbol{A}_i \, {}^{I}\boldsymbol{r} \tag{2.7} $$

defining the transformation matrix $\boldsymbol{A}_i = \boldsymbol{R}(\boldsymbol{\alpha}_i)$ as a transformation of a vector given in the global reference frame into the local reference frame. The matrix $\boldsymbol{A}_i$ is a orthonormal 3×3-matrix and hence only has 3 independent entries. Several formulations $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\alpha}_i)$ exist to calculate the transformation matrix $\boldsymbol{A}_i$ dependent on a parameterization $\boldsymbol{\alpha}_i$. A good overview on spatial rotation parameterizations is given in [61]. The three most useful ones for rotations in mechanical systems, axis-angle formulation, Euler angles and quaternions, are described in the following sections.

Axis-Angle-Formulation

A rotation in space always can be described as a rotation around an axis in space as shown in figure 2.5. It describes the rotation of a body from position 0 to position 1 around the rotation axis $\boldsymbol{d}$ by the angle φ. An arbitrary point on the body is given as $\boldsymbol{r}_0$ at the start point of the rotation and $\boldsymbol{r}_1$ at the end point of rotation.

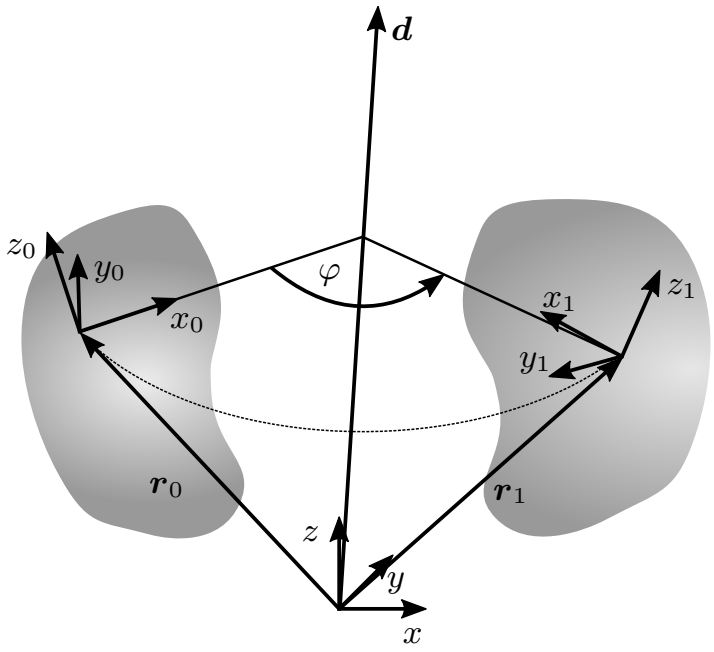


Figure 2.5: Relative axis of rotation.

According to [38, 46], transition of the vector $\boldsymbol{r}_0$ to vector $\boldsymbol{r}_1$ and the coordinate system (x_0, y_0, z_0) to the coordinate system (x_1, y_1, z_1) can be described as

$$\boldsymbol{r}_1 = \boldsymbol{r}_0 \cos(\varphi) + \boldsymbol{d}(\boldsymbol{d}^T \boldsymbol{r}_0)(1 - \cos(\varphi)) - \widetilde{\boldsymbol{d}}\,\boldsymbol{r}_0 \sin(\varphi)\,, \tag{2.8}$$

where $\widetilde{\boldsymbol{d}}$ describes the skew-symmetric matrix denoting the cross product as

$$\boldsymbol{a} \times \boldsymbol{b} = \widetilde{\boldsymbol{a}}\boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}. \tag{2.9}$$

Using the transformation of eq. (2.7) as $\boldsymbol{r}_1 = \boldsymbol{A}\boldsymbol{r}_0$, the rotation of eq. (2.8) yields the rotation matrix

$$\boldsymbol{A} = \boldsymbol{d}\boldsymbol{d}^T + (\boldsymbol{E} - \boldsymbol{d}\boldsymbol{d}^T)\cos(\varphi) - \widetilde{\boldsymbol{d}}\sin(\varphi)\,. \tag{2.10}$$

This would result in four parameters describing the rotation, three components of vector $\boldsymbol{d}$ and the rotation angle φ. The axis-angle formulation hence introduces one constraint, the length of the rotation vector

$$|\boldsymbol{d}| = 1\,. \tag{2.11}$$

Euler Angles

Since a free body in space can rotate around the three directions in space, a parametrization of the rotation matrix with rotations around these elementary directions is obvious. The rotation matrices around the x, y, z-axes can be

treated as planar rotations in the plane perpendicular to the coordinate axes and are given as

$$\boldsymbol{A}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\varphi) & \mathrm{s}(\varphi) \\ 0 & -\mathrm{s}(\varphi) & \mathrm{c}(\varphi) \end{bmatrix}; \; \boldsymbol{A}_y = \begin{bmatrix} \mathrm{c}(\theta) & 0 & -\mathrm{s}(\theta) \\ 0 & 1 & 0 \\ \mathrm{s}(\theta) & 0 & \mathrm{c}(\theta) \end{bmatrix}; \; \boldsymbol{A}_z = \begin{bmatrix} \mathrm{c}(\psi) & \mathrm{s}(\psi) & 0 \\ -\mathrm{s}(\psi) & \mathrm{c}(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

with $\mathrm{c}(\alpha) = \cos\alpha$ and $\mathrm{s}(\alpha) = \sin\alpha$.
The three elementary rotation matrices then can be combined to a rotation matrix dependent on only the three elementary rotations. However, the sequence of rotations then is crucial since the matrix product is not commutative. Rotation sequences using any combination of all the elementary rotations are often called Cardan angles or Tait-Bryan angles. Figure 2.6 shows the consecutive rotation from the (x, y, z) coordinate system to the (x''', y''', z''') coordinate system with the most common $z - y' - x''$ sequence for vehicle and aircraft dynamics.

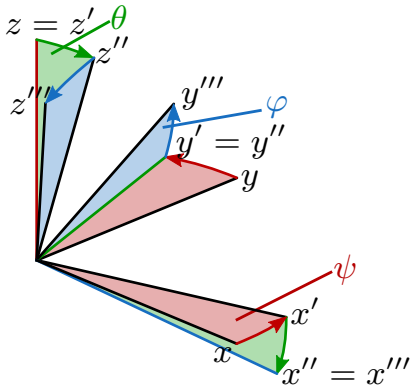


Figure 2.6: Consecutive rotations in Tait-Bryan angles, $z - y' - x''$.

For the rotation parametrization with Tait-Bryan angles $\boldsymbol{\alpha} = (\varphi, \theta, \psi)$ complete rotation matrix then reads

$$\boldsymbol{A} = \boldsymbol{A}_x \boldsymbol{A}_y \boldsymbol{A}_z = \begin{bmatrix} \mathrm{c}(\psi)\mathrm{c}(\theta) & \mathrm{c}(\theta)\mathrm{s}(\psi) & -\mathrm{s}(\theta) \\ \mathrm{c}(\psi)\mathrm{s}(\varphi)\mathrm{s}(\theta) - \mathrm{c}(\varphi)\mathrm{s}(\psi) & \mathrm{c}(\varphi)\mathrm{c}(\psi) + \mathrm{s}(\varphi)\mathrm{s}(\psi)\mathrm{s}(\theta) & \mathrm{c}(\theta)\mathrm{s}(\varphi) \\ \mathrm{s}(\varphi)\mathrm{s}(\psi) + \mathrm{c}(\varphi)\mathrm{c}(\psi)\mathrm{s}(\theta) & \mathrm{c}(\varphi)\mathrm{s}(\psi)\mathrm{s}(\theta) - \mathrm{c}(\psi)\mathrm{s}(\varphi) & \mathrm{c}(\varphi)\mathrm{c}(\theta) \end{bmatrix}. \tag{2.13}$$

Two drawbacks are obvious when looking at the complete rotation matrix given in eq. (2.13). The evaluation of the matrix requires multiple evaluation of trigonometric functions and a high number of multiplication and summation operations. This may be a limitation for efficient evaluation in numerical environments.
The back-calculation of the rotation angles from a given rotation matrix can be done by analyzing the sparsely populated entries of eq. (2.13):

$$\theta = \arcsin\left(-A_{13}\right); \quad \varphi = \arcsin\left(\frac{A_{23}}{\cos(\theta)}\right); \quad \psi = \arcsin\left(\frac{A_{12}}{\cos(\theta)}\right) \tag{2.14}$$

For $\theta = \frac{\pi}{2}$ this approach will cause a singularity since the x''- and z-axis merge to one rotation axis and all three rotation axes are located in the global $x - y$ plane. At this point, no rotation around the global z-axis is describable. An extensive discussion on Euler angle singularities, their occurrence in mechanics and their relation to the gimbal lock phenomenon can be found in [24, 27, 61].

Quaternions

Quaternions are a group of numbers that can be used to describe rotations in space as well, similar to complex numbers used to describe rotations in a plane. They were first published by Hamilton [23] and consist of four elements, one scalar, real part and three complex arguments:

$$\boldsymbol{p} = \begin{bmatrix} p_0 \\ \boldsymbol{e}_\mathrm{p} \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^T \tag{2.15}$$

This is a group of numbers in 4D-space, and by constraining the quaternions to length 1, i.e.

$$\boldsymbol{p}^T \boldsymbol{p} = 1 \tag{2.16}$$

a 3D-manifold containing the unit quaternions is generated. The four entries of a quaternion combined with one constraint equation (2.16) result in three free parameters. Defining

$$p_0 = \cos\left(\frac{\varphi}{2}\right), \tag{2.17}$$

$$\boldsymbol{e}_\mathrm{p} = \boldsymbol{d} \sin\left(\frac{\varphi}{2}\right) \tag{2.18}$$

yields a parametrization of rotations in 3D $\boldsymbol{\alpha} = \boldsymbol{p}$ that can be interpreted as the four parameters of a quaternion since they follow the same mathematical rules as quaternions. This rotation parametrization is often called Euler parameters [46] but must not be confused with the Euler angles stated above.
Inserting eq. (2.17) and eq. (2.18) in eq. (2.10) yields with trigonometric identities

$$\begin{aligned} \boldsymbol{A} &= \left(2p_0^2 - 1\right) \boldsymbol{E} + 2\left(\boldsymbol{e}_\mathrm{p}\boldsymbol{e}_\mathrm{p}^T - p_0 \widetilde{\boldsymbol{e}}_\mathrm{p}\right) \\ &= 2 \begin{bmatrix} p_0^2 + p_1^2 - \frac{1}{2} & p_0p_3 + p_1p_2 & -p_0p_2 + p_1p_3 \\ -p_0p_3 + p_1p_2 & p_0^2 + p_2^2 - \frac{1}{2} & p_0p_1 + p_2p_3 \\ p_0p_2 + p_1p_3 & -p_0p_1 + p_2p_3 & p_0^2 + p_3^2 - \frac{1}{2} \end{bmatrix}. \end{aligned} \tag{2.19}$$

Several helpful identities can be defined for quaternions, allowing an efficient and easily readable derivation of kinematics and kinetics. Many of these identities and their proofs are shown in [25, 39].
For a unit quaternion $\boldsymbol{p}$ two matrices are defined as

$$\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{p}) = \left[-\boldsymbol{e}_\mathrm{p}, p_0\boldsymbol{E} - \widetilde{\boldsymbol{e}}_\mathrm{p}\right] = \begin{bmatrix} -p_1 & p_0 & p_3 & -p_2 \\ -p_2 & -p_3 & p_0 & p_1 \\ -p_3 & p_2 & -p_1 & p_0 \end{bmatrix} \tag{2.20}$$

and

$$\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{p}) = [-\boldsymbol{e}_\mathrm{p}, p_0\boldsymbol{E} + \widetilde{\boldsymbol{e}}_\mathrm{p}] = \begin{bmatrix} -p_1 & p_0 & -p_3 & p_2 \\ -p_2 & p_3 & p_0 & -p_1 \\ -p_3 & -p_2 & p_1 & p_0 \end{bmatrix}. \tag{2.21}$$

For the $\boldsymbol{L}$-matrix, $\boldsymbol{G}$-matrix and a quaternion $\boldsymbol{p}$ the following equations hold:

$$\boldsymbol{L}\boldsymbol{p} = \boldsymbol{0} \tag{2.22}$$
$$\boldsymbol{G}\boldsymbol{p} = \boldsymbol{0} \tag{2.23}$$
$$\boldsymbol{L}\dot{\boldsymbol{p}} = -\dot{\boldsymbol{L}}\boldsymbol{p} \tag{2.24}$$
$$\dot{\boldsymbol{L}}\dot{\boldsymbol{p}} = \boldsymbol{0} \tag{2.25}$$
$$\widetilde{\boldsymbol{L}\dot{\boldsymbol{p}}} = \boldsymbol{L}\dot{\boldsymbol{L}}^T \tag{2.26}$$
$$\boldsymbol{L}\dot{\boldsymbol{L}}^T = -\dot{\boldsymbol{L}}\boldsymbol{L}^T \tag{2.27}$$
$$\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{E} - \boldsymbol{p}\boldsymbol{p}^T \tag{2.28}$$
$$\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{E} \tag{2.29}$$
$$\boldsymbol{G}\dot{\boldsymbol{L}}^T = \dot{\boldsymbol{G}}\boldsymbol{L}^T \tag{2.30}$$
$$\dot{\boldsymbol{p}} = \frac{1}{2}\boldsymbol{L}^T\boldsymbol{\omega} \tag{2.31}$$
$$\ddot{\boldsymbol{p}} = \frac{1}{2}\boldsymbol{L}^T\dot{\boldsymbol{\omega}} - \frac{1}{4}\left(\boldsymbol{\omega}^T\boldsymbol{\omega}\right)\boldsymbol{p} \tag{2.32}$$

The time-derivative of eq. (2.16) is

$$\dot{\boldsymbol{p}}^T\boldsymbol{p} = 0\,. \tag{2.33}$$

Using an arbitrary vector $\boldsymbol{e} \in \mathbb{R}^3$, two 4×4-matrices, called the *plus*- and *minus*-matrices, are defined.

$$\overset{+}{\boldsymbol{e}} = \begin{bmatrix} 0 & -\boldsymbol{e}^T \\ \boldsymbol{e} & \widetilde{\boldsymbol{e}} \end{bmatrix} \tag{2.34}$$
$$\bar{\boldsymbol{e}} = \begin{bmatrix} 0 & -\boldsymbol{e}^T \\ \boldsymbol{e} & -\widetilde{\boldsymbol{e}} \end{bmatrix} \tag{2.35}$$

For the combination of $\boldsymbol{L}$, $\boldsymbol{G}$ and $\overset{+}{\boldsymbol{e}}$, $\bar{\boldsymbol{e}}$ the following identities are given:

$$\boldsymbol{L}^T\boldsymbol{e} = \bar{\boldsymbol{e}}\boldsymbol{p} \tag{2.36}$$
$$\boldsymbol{L}\bar{\boldsymbol{e}} = -\widetilde{\boldsymbol{e}}\boldsymbol{L} + \boldsymbol{e}\boldsymbol{p}^T \tag{2.37}$$
$$\dot{\boldsymbol{L}}^T\boldsymbol{e} = \bar{\boldsymbol{e}}\dot{\boldsymbol{p}} \tag{2.38}$$
$$\ddot{\boldsymbol{L}}^T\boldsymbol{e} = \bar{\boldsymbol{e}}\ddot{\boldsymbol{p}} \tag{2.39}$$
$$\frac{\partial}{\partial \boldsymbol{p}}(\boldsymbol{A}\boldsymbol{e}) = 2\boldsymbol{L}\overset{+}{\boldsymbol{e}} + 2\boldsymbol{e}\boldsymbol{p}^T \tag{2.40}$$

$$\frac{\partial}{\partial \boldsymbol{p}}(\boldsymbol{A}^T\boldsymbol{e}) = 2\boldsymbol{G}\bar{\boldsymbol{e}} + 2\boldsymbol{e}\boldsymbol{p}^T \tag{2.41}$$

Using the $\boldsymbol{L}$-matrix and $\boldsymbol{G}$-matrix defined in eq. (2.20), eq. (2.21) the rotation matrix can be stated as

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{G}^T \ , \ \boldsymbol{A}^T = \boldsymbol{G}\boldsymbol{L}^T \ . \tag{2.42}$$

The relationships (2.22)–(2.42) can be verified by inserting eq. (2.15), (2.20) and (2.21) and component wise calculation.
One advantage of quaternions is the relatively simple assembly of the rotation matrix and, as will be shown later in this chapter, the straightforward sensitivity analysis for constraint equations. In contrast to Tait-Bryan-angles the back-calculation of quaternion elements does not result in singularities. This can be done by evaluating the trace of the rotation matrix

$$\begin{aligned} \mathrm{tr}(\boldsymbol{A}) &= A_{11} + A_{22} + A_{33} \\ &= 6p_0^2 + 2p_1^2 + 2p_2^2 + 2p_3^2 - 3 \\ &= 4p_0^2 - 1 \\ \Rightarrow p_0 &= \sqrt{\frac{\mathrm{tr}(\boldsymbol{A}) + 1}{4}} \end{aligned} \tag{2.43}$$

and re-insertion into the diagonal elements of $\boldsymbol{A}$:

$$p_1 = \frac{1}{4p_0}\left(A_{23} - A_{32}\right) \tag{2.44}$$

$$p_2 = \frac{1}{4p_0}\left(A_{31} - A_{13}\right) \tag{2.45}$$

$$p_3 = \frac{1}{4p_0}\left(A_{12} - A_{21}\right). \tag{2.46}$$

The trace of the matrix may reach values of $\mathrm{tr}(\boldsymbol{A}) = -1$ for some rotations and hence $p_0 = 0$. Thus, eq. (2.44) – eq. (2.46) cannot be solved. This problem can be circumvented by evaluation of another element of the quaternion first. For example, for p_1 follows

$$\begin{aligned} A_{11} - A_{22} - A_{33} &= -2p_0^2 + 2p_1^2 - 2p_2^2 - 2p_3^2 + 1 \\ &= 4p_1^2 - 1 \\ \Rightarrow p_1 &= \sqrt{\frac{A_{11} - A_{22} - A_{33} + 1}{4}} \end{aligned} \tag{2.47}$$

and similar for p_2 and p_3

$$p_2 = \sqrt{\frac{A_{22} - A_{11} - A_{33} + 1}{4}} \tag{2.48}$$

$$p_3 = \sqrt{\frac{A_{33} - A_{22} - A_{11} + 1}{4}} \, . \tag{2.49}$$

Depending on the values of the diagonal elements there exists always a set of quaternion parameters describing the rotation of matrix $\boldsymbol{A}$. Since the signs of eq. (2.43) and eq. (2.47), eq. (2.48), eq. (2.49) are not specified, there are always two quaternions describing the same rotation, similar to the free sign choice for the axis-angle description.

2.1.4 Angular Velocity

While the calculation of translational velocity components is straight-forward, the kinematic analysis of rotations requires further investigation since angular movement is represented by a coordinate transformation as shown in eq. (2.7). Consider a local vector $\boldsymbol{r}_k$ on a rigid body as shown in figure (2.4). In global coordinates this vector is written as

$$ {}^{I}\boldsymbol{r}_k = \boldsymbol{A}_i^T \, {}^{i}\boldsymbol{r}_k \tag{2.50} $$

with the trailing upper index denoting the coordinate system used to express the vector. Taking the time derivative of eq. (2.50) with respect to the inertial frame gives

$$ \frac{\mathrm{d}}{\mathrm{d}t} {}^{I}\boldsymbol{r}_k = \left(\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{A}_i^T \right) {}^{i}\boldsymbol{r}_k + \underbrace{\boldsymbol{A}_i^T \frac{\mathrm{d}}{\mathrm{d}t} {}^{i}\boldsymbol{r}_k}_{=0, \boldsymbol{r}_k \text{ fixed}} = \dot{\boldsymbol{A}}_i^T \, {}^{i}\boldsymbol{r}_k \tag{2.51} $$

and with eq. (2.7) to express all equations in the global reference frame

$$ \frac{\mathrm{d}}{\mathrm{d}t} {}^{I}\boldsymbol{r}_k = \underbrace{\dot{\boldsymbol{A}}_i^T \boldsymbol{A}_i}_{= {}^{I}\widetilde{\boldsymbol{\omega}}_i} {}^{I}\boldsymbol{r}_k \tag{2.52} $$

the global angular velocity tensor ${}^{I}\widetilde{\boldsymbol{\omega}}_i$ is defined. The tensor defines the relation between the vector ${}^{I}\boldsymbol{r}_k$ and its time derivative in global coordinates. It is a skew-symmetric tensor of the structure shown in eq. (2.9) and eq. (2.52) is equal to

$$ \frac{\mathrm{d}}{\mathrm{d}t} {}^{I}\boldsymbol{r}_k = {}^{I}\dot{\boldsymbol{r}}_k = {}^{I}\widetilde{\boldsymbol{\omega}}_i \, {}^{I}\boldsymbol{r}_k = {}^{I}\boldsymbol{\omega}_i \times {}^{I}\boldsymbol{r}_k \, . \tag{2.53} $$

The angular velocity vector and tensor can be transformed into the local or body reference frame and reads as

$$ {}^{i}\boldsymbol{\omega}_i = \boldsymbol{A}_i \, {}^{I}\boldsymbol{\omega}_i \tag{2.54} $$

$$ {}^{i}\widetilde{\boldsymbol{\omega}}_i = \boldsymbol{A}_i \, {}^{I}\widetilde{\boldsymbol{\omega}}_i \boldsymbol{A}_i^T = \boldsymbol{A}_i \dot{\boldsymbol{A}}_i^T \tag{2.55} $$

For all successive equations of this chapter the representation of the angular velocity in the local reference frame is used. Similar calculations can be conducted for the global reference frame.

Angular Velocity for Quaternion Parametrization

For a rotation parametrization with unit quaternions the time-derivative of the rotation matrix can be calculated by deriving eq. (2.42) and using identity eq. (2.30):

$$\begin{aligned}\dot{\boldsymbol{A}} &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}\boldsymbol{G}^T = \left(\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}\right)\boldsymbol{G}^T + \boldsymbol{L}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{G}^T\right) \\ &= \dot{\boldsymbol{L}}\boldsymbol{G}^T + \boldsymbol{L}\dot{\boldsymbol{G}}^T \\ &= 2\dot{\boldsymbol{L}}\boldsymbol{G}^T \end{aligned} \tag{2.56}$$

For eq. (2.55) then follows with eq. (2.56), eq. (2.22), eq. (2.26), eq. (2.28)

$$\begin{aligned}\widetilde{\boldsymbol{\omega}} = \boldsymbol{A}\dot{\boldsymbol{A}}^T &= \boldsymbol{L}\boldsymbol{G}^T\left(2\dot{\boldsymbol{L}}\boldsymbol{G}^T\right)^T \\ &= 2\boldsymbol{L}\boldsymbol{G}^T\boldsymbol{G}\dot{\boldsymbol{L}}^T \\ &= 2\boldsymbol{L}\left(\boldsymbol{E} - \boldsymbol{p}\boldsymbol{p}^T\right)\dot{\boldsymbol{L}}^T \\ &= 2\boldsymbol{L}\dot{\boldsymbol{L}}^T - 2\underbrace{\boldsymbol{L}\boldsymbol{p}}_{=\boldsymbol{0}}\boldsymbol{p}^T\dot{\boldsymbol{L}}^T \\ &= 2\widetilde{\boldsymbol{L}\dot{\boldsymbol{p}}}. \\ \Rightarrow\ \boldsymbol{\omega} &= 2\boldsymbol{L}\dot{\boldsymbol{p}}\end{aligned} \tag{2.57}$$

By left-multiplication of eq. (2.57) with $\boldsymbol{L}^T$ and using identities eq. (2.28), eq. (2.33) the equation can be inverted and results in eq. (2.31).

$$\begin{aligned}\boldsymbol{L}^T\boldsymbol{\omega} &= 2\boldsymbol{L}^T\boldsymbol{L}\dot{\boldsymbol{p}} \\ &= 2\left(\boldsymbol{E} - \boldsymbol{p}\boldsymbol{p}^T\right)\dot{\boldsymbol{p}} \\ &= 2\dot{\boldsymbol{p}} \\ \Rightarrow \dot{\boldsymbol{p}} &= \frac{1}{2}\boldsymbol{L}^T\boldsymbol{\omega}\end{aligned} \tag{2.58}$$

The kinematic relationship between $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ hence is given as

$$\dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{p}} = \frac{1}{2}\boldsymbol{L}^T\boldsymbol{\omega} = \boldsymbol{Z}_r\left(\boldsymbol{\alpha}\right)\boldsymbol{\omega} \tag{2.59}$$

with the matrix $\boldsymbol{Z}_r\left(\boldsymbol{\alpha}\right)$ called kinematic matrix of the parametrization $\boldsymbol{\alpha}$.

Angular Velocity for Tait-Bryan Angles

Using Tait-Bryan Angles to parametrize the rotation, similar relationships to eq. (2.57) and eq. (2.31) can be found as

$$\begin{aligned}\boldsymbol{\omega} &= \begin{bmatrix}\dot{\varphi}\\0\\0\end{bmatrix} + \begin{bmatrix}1 & 0 & 0\\0 & \cos(\varphi) & \sin(\varphi)\\0 & -\sin(\varphi) & \cos(\varphi)\end{bmatrix}\left\{\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \begin{bmatrix}\cos(\theta) & 0 & -\sin(\theta)\\0 & 1 & 0\\\sin(\theta) & 0 & \cos(\theta)\end{bmatrix}\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}\right\}\\ \boldsymbol{\omega} &= \begin{bmatrix}1 & 0 & -\sin(\theta)\\0 & \cos(\varphi) & \cos(\theta)\sin(\varphi)\\0 & -\sin(\varphi) & \cos(\varphi)\cos(\theta)\end{bmatrix}\begin{bmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}. \end{aligned} \tag{2.60}$$

Inverting eq. (2.60) results in the kinematic matrix

$$\dot{\boldsymbol{\alpha}} = \begin{bmatrix}\dot{\varphi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix} = \boldsymbol{Z}_r(\boldsymbol{\alpha})\,\boldsymbol{\omega} = \begin{bmatrix}1 & \dfrac{\sin(\varphi)\sin(\theta)}{\cos(\theta)} & \dfrac{\cos(\varphi)\sin(\theta)}{\cos(\theta)}\\0 & \cos(\varphi) & -\sin(\varphi)\\0 & \dfrac{\sin(\varphi)}{\cos(\theta)} & \dfrac{\cos(\varphi)}{\cos(\theta)}\end{bmatrix}\boldsymbol{\omega} \tag{2.61}$$

and shows again the drawback of having singularities at $\cos\left(\pm\frac{\pi}{2}, \pm\frac{3\pi}{2}, \dots\right)$ for Tait-Bryan angles.

2.1.5 Kinetics of a Rigid Body

Forces and moments acting on mass points and on rigid bodies result in a change of the body motion. The simple case of a single force acting on a mass point as shown in figure 2.2 results in an acceleration of the point, this is stated by Newton's second law as

$$m_i \boldsymbol{a}_i = \boldsymbol{f}_i \tag{2.62}$$

and is valid for the translational movement in all three directions. A mass point does not have any spatial extension and thus the force or the sum of all forces $\boldsymbol{f}_i$ acting on the body result only in a translational movement. Newton's second law can be formulated in any cartesian reference frame, however, for moving reference frames the relative motion has to be taken into account. The above version is only valid for the absolute acceleration.
For purely translational motion and forces acting on the center of gravity Newton's second law will fully describe the motion of a rigid body. If forces act on a body offset to the center of gravity or for moments acting on the body, this will result in additional rotational motion of the body. This is describes by Euler's equation, which is stated as

$${}^{i}\boldsymbol{I}_i\,{}^{i}\dot{\boldsymbol{\omega}}_i + {}^{i}\tilde{\boldsymbol{\omega}}\,{}^{i}\boldsymbol{I}_i\,{}^{i}\boldsymbol{\omega}_i = {}^{i}\boldsymbol{\ell}_i\ . \tag{2.63}$$

Hereby ${}^i\boldsymbol{\ell}_i$ states the sum of all external moments acting on the body. It includes free moments ${}^i\boldsymbol{\ell}_{f,i}$ and the moment of all forces ${}^i\boldsymbol{f}_k$ applied on the body at the force application points ${}^i\boldsymbol{r}_k$ with

$$ {}^i\boldsymbol{\ell}_i = \sum {}^i\boldsymbol{\ell}_{f,i} + \sum {}^i\widetilde{\boldsymbol{r}}_k \, {}^i\boldsymbol{f}_k \tag{2.64} $$

The rotational inertia of the body is described by the inertia tensor $\boldsymbol{I}_i$. It can be stated as an integral over the continuum Ω_i describing body i as

$$ {}^i\boldsymbol{I}_i = \int_{\Omega_i} {}^i\widetilde{\boldsymbol{r}}_k \, {}^i\widetilde{\boldsymbol{r}}_k^T \, \mathrm{d}m \ . \tag{2.65} $$

Expressed in the body-fixed reference frame K_i, the inertia tensor only depends on the material coordinates. Hence, since the material coordinates do not change in rigid bodies, the inertia tensor is constant for rigid bodies. Euler's equation can be stated in the global reference frame K_I as well, resulting in

$$ \begin{aligned} \boldsymbol{A}_i^T \, {}^i\boldsymbol{\ell}_i &= \boldsymbol{A}_i^T \, {}^i\boldsymbol{I}_i \boldsymbol{A}_i \, {}^I\dot{\boldsymbol{\omega}}_i + {}^I\widetilde{\boldsymbol{\omega}} \boldsymbol{A}_i^T \, {}^i\boldsymbol{I}_i \boldsymbol{A}_i \, {}^I\boldsymbol{\omega}_i \\ {}^I\boldsymbol{\ell}_i &= {}^I\boldsymbol{I}_i \, {}^I\dot{\boldsymbol{\omega}}_i + {}^I\widetilde{\boldsymbol{\omega}} \, {}^I\boldsymbol{I}_i \, {}^I\boldsymbol{\omega}_i \ . \end{aligned} \tag{2.66} $$

Summarized from the calculations above the EOM for a free rigid body in space are

$$ m_i \boldsymbol{a}_i = \boldsymbol{f}_i \tag{2.67} $$

$$ {}^i\boldsymbol{I}_i \, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\widetilde{\boldsymbol{\omega}} \, {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i = {}^i\boldsymbol{\ell}_i \ . \tag{2.68} $$

2.1.6 State-Space Formulation of Rigid Body Systems

The EOM stated in eq. (2.67) and eq. (2.68) form a system of second order differential equations. The states of a body i are represented by the position and velocity variables, they can be written in vector form as

$$ \boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{\rho}_i \\ \boldsymbol{\alpha}_i \end{bmatrix} \tag{2.69} $$

$$ \boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \ . \tag{2.70} $$

The angular velocity $\boldsymbol{\omega}_i$ is related to the rotation parametrization $\boldsymbol{\alpha}_i$ with the kinematic matrix $\boldsymbol{Z}_r(\boldsymbol{\alpha}_i)$, which is itself dependent on the rotation parametrization $\boldsymbol{\alpha}_i$. For $\boldsymbol{\rho}_i$, $\boldsymbol{v}_i$ defined in the global reference frame follows

$$ \begin{aligned} \begin{bmatrix} \dot{\boldsymbol{\rho}}_i \\ \dot{\boldsymbol{\alpha}}_i \end{bmatrix} &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_r(\boldsymbol{\alpha}_i) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \\ \dot{\boldsymbol{x}}_i &= \boldsymbol{Z}(\boldsymbol{\alpha}_i) \, \boldsymbol{z}_i \end{aligned} \tag{2.71} $$

and Newton and Euler's equations can be written as a system of first order differential equations as

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \boldsymbol{Z}\left(\boldsymbol{\alpha}_i\right) \boldsymbol{z}_i \\ \begin{bmatrix} m_i \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & {}^i\boldsymbol{I}_i \end{bmatrix} \dot{\boldsymbol{z}}_i &= \begin{bmatrix} \boldsymbol{f}_i \\ {}^i\boldsymbol{\ell}_i - {}^i\widetilde{\boldsymbol{\omega}}\, {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i \end{bmatrix} . \end{aligned} \tag{2.72}$$

2.1.7 Constrained Multibody Systems

For an unconstrained multibody system the EOM can be stated for each body separately, the state variables of the bodies are independent of each other. In most multibody systems the bodies are connected by joints such as ball bearings, rails, gears or connection rods. This reduces the total number of DOF of the system. Although most joints show a small compliance, modeling a multibody system with stiff spring-damper forces and moments acting on the bodies will result in a system of EOM with high numerical stiffness and is usually not an option for efficient simulation [45].
The movement of bodies in multibody systems relative to the environment or to other bodies can be limited by introducing algebraic equations. In figure 2.3 the constraint of a mass point to a curve in space is shown. The corresponding constraint equation for this constraint would read

$$\boldsymbol{g}\left(\boldsymbol{\rho}_i\right) = \boldsymbol{\rho}_i - \boldsymbol{r}_s = \boldsymbol{0} \tag{2.73}$$

with $\boldsymbol{r}_s$ denoting a parametrization of the curve s in space. The constraint equation is dependent of the position variable $\boldsymbol{\rho}_i$ of body i, so it is called a holonomic constraint. Constraint equations stated on velocity level which are not integrable are called nonholonomic. These constraints are not covered in this work. The constraint can as well be dependent on time, called a rheonomic constraint. Constraints independent on time are called skleronomic.
A generalized description of a rheonomic constraint between two bodies i, j is

$$g\left(\boldsymbol{x}_i, \boldsymbol{x}_j, t\right) = 0 \ . \tag{2.74}$$

This is an implicit description since the equation is not resolved explicitly to $\boldsymbol{\rho}_i$ or $\boldsymbol{\rho}_j$. Every constraint equation added to the system of EOM is reducing the number of DOF by one if it is linearly independent of all other constraint equations.
Constraint equations can be stated either in the global reference frame K_I, in a local reference frame K_i or joint-related coordinate system K_k. For mass point systems and rigid body systems the global coordinate system will produce quite simple equations that simplify a sensitivity analysis. Multibody systems considering body flexibilities can be described in joint-related coordinates [57] or in global coordinates as well.
The two most common joints are described subsequently, a ball joint or spherical joint and a rotational joint, see figure 2.7 and figure 2.8. Many other joints

equations can be derived similar to these two joints. Planar problems can be solved by considering only one or two rows of the spatial problems.

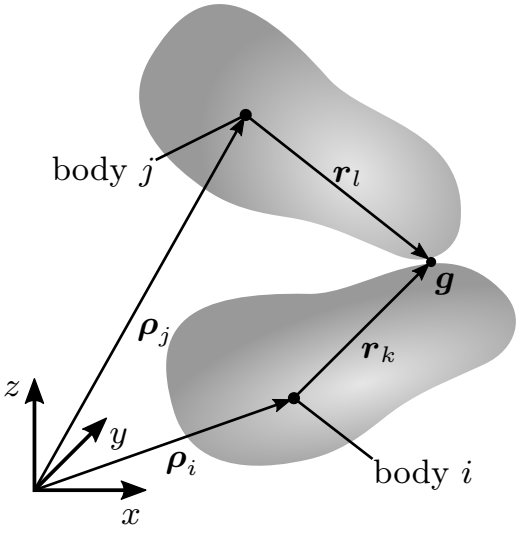


Figure 2.7: Ball joint.

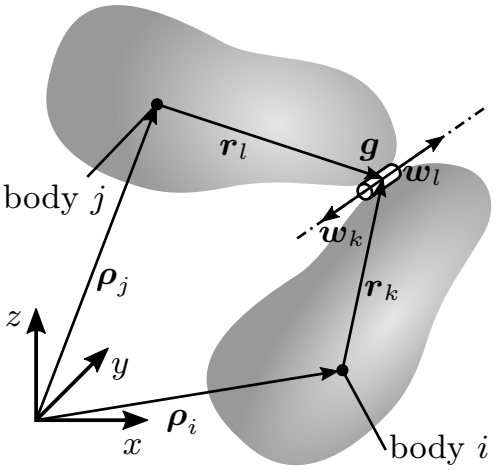


Figure 2.8: Rotational joint.

Ball Joint

A ball joint is connecting two bodies by requiring that two points of the constrained bodies coincide. Thus all relative translational motion is blocked while all three relative rotational motion capabilities between the bodies are maintained. In global coordinates this can be written as

$$\boldsymbol{g} = \boldsymbol{\rho}_i + \boldsymbol{A}_i^T \boldsymbol{r}_k - \boldsymbol{\rho}_j - \boldsymbol{A}_j^T \boldsymbol{r}_l = \boldsymbol{0} \ . \tag{2.75}$$

The equation can be stated in local coordinates of one of the bodies as well. For example, in coordinates of body j this yields

$$\boldsymbol{g} = \boldsymbol{A}_j \left(\boldsymbol{\rho}_i + \boldsymbol{A}_i^T \boldsymbol{r}_k - \boldsymbol{\rho}_j \right) - \boldsymbol{r}_l = \boldsymbol{0} \ . \tag{2.76}$$

By considering only one or two rows of the equation this constraint allows addition to only one or two local translations. Introducing additional local rotations at the points k, l can be used to tilt the bodies relative to each other or, in case of flexible bodies, allow rotational deformation of the body. When all three constraint equations are considered, the constraint removes three DOF from the multibody system.

Rotational Joint

Rotational joints restrict the relative movement of two bodies to only one rotational DOF, hence removing 5 DOF from the multibody system. Since no translational movement is allowed, a rotational joint includes the constraint equations

of a ball joint. Two options exist to define the additional limitation of rotational movement. The first option is to calculate the local joint rotation matrix as

$$\boldsymbol{A}_c = \boldsymbol{A}_j \boldsymbol{A}_i^T \tag{2.77}$$

and then recalculate the values $\boldsymbol{\alpha}_c$ of a rotation parametrization $\boldsymbol{A}_c = \boldsymbol{A}_c(\boldsymbol{\alpha}_c)$. Since the constraint is supposed to allow the rotation about only one axis of the joint, it is possible to use Euler angles as a local rotation parametrization and setting two out of the three angles of $\boldsymbol{\alpha}_c$ to zero, though some rotation sequences still can cause singular configurations. In the case of a local rotation that defines the relative orientation of the nodal coordinate systems K_k, K_l to the body-related system K_i the joint rotation matrix extends to

$$\boldsymbol{A}_c = \boldsymbol{B}_l \boldsymbol{A}_j \boldsymbol{A}_i^T \boldsymbol{B}_k^T \ . \tag{2.78}$$

While this definition of the constraint is quite straightforward, its drawback is obvious: The back-calculation of the rotation parameterization requires high calculation effort and could lead to singularities for some choices of the parametrization.
The second option is the definition of two vectors $\boldsymbol{w}_k$ and $\boldsymbol{w}_l$ representing the rotation axes at point k and point l. The requirement to keep these vectors parallel can be stated as

$$\boldsymbol{g} = {}^I\widetilde{\boldsymbol{w}}_k \, {}^I\boldsymbol{w}_l = \boldsymbol{0} \ . \tag{2.79}$$

It is worth mentioning that for this definition the vectors have to be defined in the same reference system. This definition of the constraint is relatively simple, but results in three equations for only two constrained DOF. So one equation is always redundant and the two linearly independent equations have to be selected, whereby the selection of the independent constraint equation depends on the orientation of the bodies [38].

2.1.8 Principles of Mechanics

Newton's and Euler's EOM stated in eq. (2.67) and eq. (2.68) are valid for free systems with forces and moments explicitly calculable from position and velocity of the bodies. However, introducing algebraic constraints adds constraint forces to the system that are unknown in advance, the equations are then extended to

$$m_i \boldsymbol{a}_i = \boldsymbol{f}_i^a + \boldsymbol{f}_i^c \tag{2.80}$$

$${}^i\boldsymbol{I}_i \, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\widetilde{\boldsymbol{\omega}} \, {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i = {}^i\boldsymbol{\ell}_i^a + {}^i\boldsymbol{\ell}_i^c \tag{2.81}$$

with $\boldsymbol{f}_i^a$, ${}^i\boldsymbol{\ell}_i^a$ stating the applied forces and moments and $\boldsymbol{f}_i^c$, ${}^i\boldsymbol{\ell}_i^c$ stating the constraint forces and moments acting on the bodies of the MBS.
Two principles, D'Alembert's principle of virtual work and Jourdain's principle of virtual power are employed to either calculate or eliminate the unknown constraint forces. A comprehensive derivation of these principles can be found in [49].

D'Alembert's Principle

In figure 2.9 a mass point is constrained to a line. The constraint forces $\boldsymbol{f}_i^c$, which are unknown in advance, ensure that the mass point stays on the line.

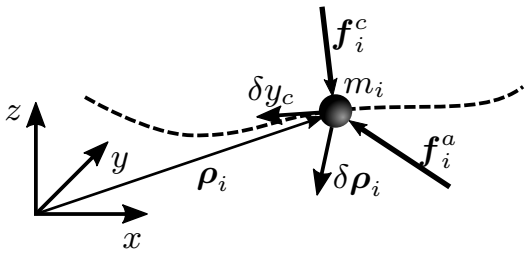


Figure 2.9: Mass point, virtual displacement and constraint force.

Introducing an infinitesimally small virtual displacement $\delta\boldsymbol{\rho}_i$ for each body i that is compatible with the constraints of the mass point, the virtual work conducted by the constraint forces equals zero,

$$\delta W = \boldsymbol{f}_i^{cT} \delta\boldsymbol{\rho}_i = 0 \ . \tag{2.82}$$

For multibody systems with n bodies, the virtual work for the sum of all constraint forces vanishes and by inserting Newton's equation from eq. (2.80) in eq. (2.82) results in D'Alembert's principle in the formulation of Lagrange,

$$\delta W = \sum_{i=1}^{n} \left(m_i \boldsymbol{a}_i - \boldsymbol{f}_i^a\right)^T \delta\boldsymbol{\rho}_i = 0 \ . \tag{2.83}$$

For systems of point masses and rigid bodies follows with virtual, infinitesimal rotations $\delta\boldsymbol{s}_i$ in the same way

$$\delta W = \sum_{i=1}^{n} \left(m_i \boldsymbol{a}_i - \boldsymbol{f}_i^a\right)^T \delta\boldsymbol{\rho}_i + \sum_i \left({}^i\boldsymbol{I}_i\, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\tilde{\boldsymbol{\omega}}_i\, {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i - {}^i\boldsymbol{\ell}_i^a\right)^T \delta\boldsymbol{s}_i = 0 \ . \tag{2.84}$$

It is important that $\delta\boldsymbol{\rho}_i, \delta\boldsymbol{s}_i$ are compatible with all constraints of the MBS, which states that the variations of $\boldsymbol{\rho}_i, \boldsymbol{s}_i$ are not independent for constrained systems.

Jourdain's Principle

Similar to D'Alembert's principle, Jourdain's principle can be derived. The principle of virtual power states that the virtual power of the reaction forces equals zero. Therefore, virtual velocity variations $\delta'\boldsymbol{v}_i$, $\delta'\boldsymbol{\omega}_i$ are introduced, they must be compatible with the constraint equations. Replacing the position

variations of eq. (2.84), it follows

$$\begin{aligned}\delta' P = & \sum_{i=1}^{n} \delta' \boldsymbol{v}_i^T \left(m_i \boldsymbol{a}_i - \boldsymbol{f}_i^a \right) \\ & + \sum_{i=1}^{n} \delta' \boldsymbol{\omega}_i^T \left({}^i\boldsymbol{I}_i \, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\tilde{\boldsymbol{\omega}}_i \, {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i - {}^i\boldsymbol{\ell}_i^a \right) = 0 \,. \end{aligned} \tag{2.85}$$

Variation of Constraint Equations

The variations $\delta\boldsymbol{\rho}_i, \delta\boldsymbol{s}_i$ of D'Alembert's principle and $\delta'\boldsymbol{v}_i, \delta'\boldsymbol{\omega}_i$ of Jourdain's principle are only independent for unconstrained systems. For any constrained mechanical system, a set of m algebraic equations

$$g_m \left(\boldsymbol{\rho}, \boldsymbol{\alpha} \right) = 0 \tag{2.86}$$

arises, creating dependencies among the variations $\delta\boldsymbol{\rho}_i, \delta\boldsymbol{s}_i$. These follow from the variation of the constraint equations

$$\delta g_m = \frac{\partial g_m}{\partial \boldsymbol{\rho}_i} \delta\boldsymbol{\rho}_i = \boldsymbol{G}_{\boldsymbol{\rho}_i} \left(\boldsymbol{\rho}, \boldsymbol{\alpha} \right) \delta\boldsymbol{\rho}_i = 0 \tag{2.87}$$

and similarly for the time derivative of the constraint equations

$$\delta' \dot{g}_m = \frac{\partial g_m}{\partial \boldsymbol{\rho}_i} \delta' \dot{\boldsymbol{\rho}}_i = \boldsymbol{G}_{\boldsymbol{\rho}_i} \left(\boldsymbol{\rho}, \boldsymbol{\alpha} \right) \delta' \boldsymbol{v}_i = 0 \tag{2.88}$$

with $\delta'\dot{\boldsymbol{\rho}}_i = \delta'\boldsymbol{v}_i$.
For a constraint variational problem

$$\begin{aligned} \boldsymbol{c}^T \delta\boldsymbol{\rho} = 0 \,, \qquad & \forall \delta\boldsymbol{\rho} : \ \boldsymbol{G}_{\boldsymbol{\rho}} \delta\boldsymbol{\rho} = 0 \quad \text{and} \\ \boldsymbol{c}^T \delta'\boldsymbol{v} = 0 \,, \qquad & \forall \delta'\boldsymbol{v} : \ \boldsymbol{G}_{\boldsymbol{\rho}} \delta'\boldsymbol{v} = 0 \end{aligned} \tag{2.89}$$

a set of Lagrange multipliers $\boldsymbol{\lambda}$ can be introduced such that

$$\begin{aligned} \left(\boldsymbol{c}^T + \boldsymbol{\lambda}^T \boldsymbol{G}_{\boldsymbol{\rho}} \right) \delta\boldsymbol{\rho} = 0 \quad & \text{and} \\ \left(\boldsymbol{c}^T + \boldsymbol{\lambda}^T \boldsymbol{G}_{\boldsymbol{\rho}} \right) \delta'\boldsymbol{v} = 0 \,, \end{aligned} \tag{2.90}$$

see [7, 25]. Similar calculations can be performed for $\delta\boldsymbol{s}, \delta'\dot{\boldsymbol{s}}$ and eq. (2.87) – eq. (2.90) by considering the kinematic relationship of eq. (2.59) with

$$\delta\boldsymbol{s}_i = \boldsymbol{Z}_r \left(\boldsymbol{\alpha}_i \right) \delta\boldsymbol{\omega}_i \,. \tag{2.91}$$

Inserting eq. (2.85) in eq. (2.90) yields

$$\begin{aligned}\delta' P = & \sum_{i} \delta' \boldsymbol{v}_i^T \left(m_i \boldsymbol{a}_i - \boldsymbol{f}_i^a + \boldsymbol{G}_{\boldsymbol{\rho}_i}^T \boldsymbol{\lambda} \right) \\ & + \sum_{i} \delta' \boldsymbol{\omega}_i^T \left({}^i\boldsymbol{I}_i \, {}^i\dot{\boldsymbol{\omega}}_i + {}^i\tilde{\boldsymbol{\omega}}_i \, {}^i\boldsymbol{I}_i \, {}^i\boldsymbol{\omega}_i - {}^i\boldsymbol{\ell}_i^a + \boldsymbol{G}_{\boldsymbol{s}_i}^T \boldsymbol{\lambda} \right) = 0 \,. \end{aligned} \tag{2.92}$$

which can be transformed into a system of $2n$ coupled differential equations as

$$\dot{\boldsymbol{x}} = \boldsymbol{Z}\boldsymbol{z} \tag{2.93}$$

$$\boldsymbol{M}\dot{\boldsymbol{z}} = \boldsymbol{f}^a - \boldsymbol{G}^T\boldsymbol{\lambda} \tag{2.94}$$

with eq. (2.93) representing the kinematic equations to formulate a set of first order differential equations. Equation (2.93) can be included in eq. (2.94) to create a second order system of n coupled equations. The other elements are combined of the elements of all bodies i and constraints m as

$$\begin{aligned}
\boldsymbol{x} &= \left[\boldsymbol{\rho}_1,\ \dot{\boldsymbol{\alpha}}_1,\ \cdots,\ \boldsymbol{\rho}_n,\ \dot{\boldsymbol{\alpha}}_n\right]^T \\
\boldsymbol{z} &= \left[\boldsymbol{v}_1,\ {}^i\boldsymbol{\omega}_1,\ \cdots,\ \boldsymbol{v}_n,\ {}^i\boldsymbol{\omega}_n\right]^T \\
\boldsymbol{Z} &= \begin{bmatrix} \boldsymbol{E} & & & & \boldsymbol{0} \\ & \boldsymbol{Z}_r\left(\boldsymbol{\alpha}_1\right) & & & \\ & & \ddots & & \\ & & & \boldsymbol{E} & \\ \boldsymbol{0} & & & & \boldsymbol{Z}_r\left(\boldsymbol{\alpha}_n\right) \end{bmatrix} \\
\boldsymbol{M} &= \begin{bmatrix} m_1\boldsymbol{E} & & & & \boldsymbol{0} \\ & {}^i\boldsymbol{I}_1 & & & \\ & & \ddots & & \\ & & & m_n\boldsymbol{E} & \\ \boldsymbol{0} & & & & {}^i\boldsymbol{I}_n \end{bmatrix} \\
\boldsymbol{f}^a &= \left[\boldsymbol{f}_1^a,\ -{}^i\tilde{\boldsymbol{\omega}}_1\,{}^i\boldsymbol{I}_1\,{}^i\boldsymbol{\omega}_1 + {}^i\boldsymbol{\ell}_1^a,\ \cdots,\ \boldsymbol{f}_n^a,\ -{}^i\tilde{\boldsymbol{\omega}}_n\,{}^i\boldsymbol{I}_n\,{}^i\boldsymbol{\omega}_n + {}^i\boldsymbol{\ell}_n^a\right]^T \\
\boldsymbol{G} &= \begin{bmatrix} \boldsymbol{G}_{\boldsymbol{\rho}_1}^1 & \boldsymbol{G}_{\boldsymbol{s}_1}^1 & \cdots & \boldsymbol{G}_{\boldsymbol{\rho}_n}^1 & \boldsymbol{G}_{\boldsymbol{s}_n}^1 \\ \vdots & & & & \vdots \\ \boldsymbol{G}_{\boldsymbol{\rho}_1}^m & \boldsymbol{G}_{\boldsymbol{s}_1}^m & \cdots & \boldsymbol{G}_{\boldsymbol{\rho}_n}^m & \boldsymbol{G}_{\boldsymbol{s}_n}^m \end{bmatrix}.
\end{aligned} \tag{2.95}$$

Adding the constraint equations $\boldsymbol{g}\left(\boldsymbol{\rho}, \boldsymbol{\alpha}\right)$ finally yields *Lagrange's equations of 1st kind*

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{Z}\boldsymbol{z} \\
\boldsymbol{M}\dot{\boldsymbol{z}} &= \boldsymbol{f}^a - \boldsymbol{G}^T\boldsymbol{\lambda} \\
\boldsymbol{g}\left(\boldsymbol{\rho}, \boldsymbol{\alpha}\right) &= \boldsymbol{0}
\end{aligned} \tag{2.96}$$

which are also called descriptor form. These are a set of $6\,n + m$ DAEs for a system of only $6\,n - m$ DOF. The main drawback of these equations is the large system dimension for, depending on the number of constraints, low number of DOF. This increases the complexity for numerical integration of the equations. It may be advantageous for numerical integration that the matrices $\boldsymbol{Z}$ and $\boldsymbol{M}$ are sparse and that the applied forces are formulated with respect to the cartesian coordinates of the bodies.

2.1.9 Formulation in Generalized Coordinates

In eq. (2.90) dependent variations of $\delta' \boldsymbol{v}_i$ and $\delta' \boldsymbol{\omega}_i$ are used to gain a set of EOM compliant with the constraint equations, called *Lagrange's equations of first kind*.
Depending on the structure of the multibody system, it is possible to introduce a set of generalized coordinates that is compliant with the constraint equations. Hereby the number of coordinates corresponds to the number of DOF. Figure 2.10 shows the two general occurring topologies. While there is always a unique path to describe all bodies of a tree-like structure, two different paths are possible to describe a body in a closed, kinematic loop. For tree-like structures a unique set of generalized coordinates can always be found e.g. by using relative coordinates describing the relative motion between the constrained bodies. For closed loops this is not possible in general, however, for some common problems like vehicle suspensions analytic solutions of the kinematic loops have been found [44].

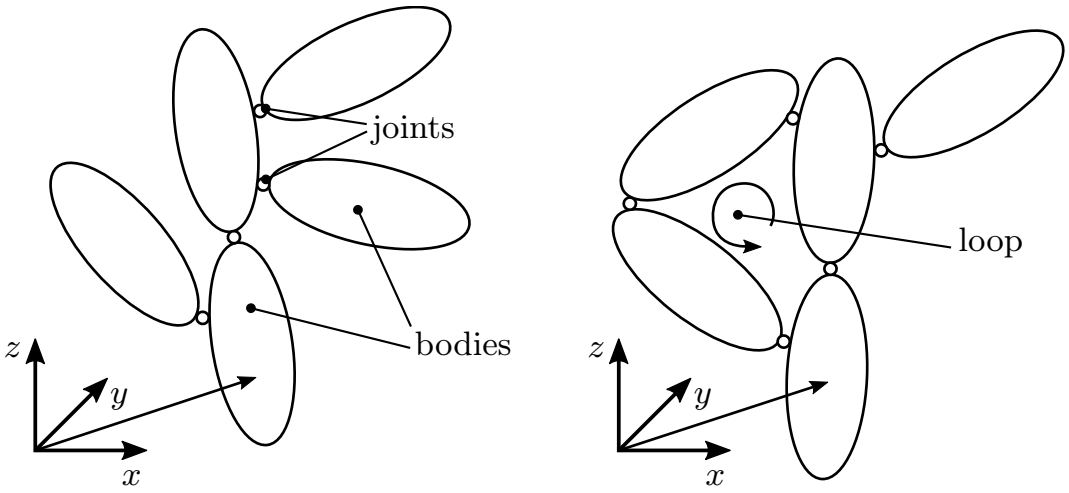


Figure 2.10: Structures of multibody systems: Tree (left) and loop (right).

Introducing a set of generalized position coordinates $\boldsymbol{y}$ and velocity coordinates $\dot{\boldsymbol{y}}$ yields the position variations $\delta \boldsymbol{y}$ and velocity variations $\delta' \dot{\boldsymbol{y}}$ with

$$\begin{aligned} \delta \boldsymbol{\rho}_i &= \frac{\partial \boldsymbol{\rho}_i}{\partial \boldsymbol{y}} \delta \boldsymbol{y} = \boldsymbol{J}_i^t \, \delta \boldsymbol{y} & \qquad \delta \boldsymbol{s}_i &= \frac{\partial \boldsymbol{s}_i}{\partial \boldsymbol{y}} \delta \boldsymbol{y} = \boldsymbol{J}_i^r \, \delta \boldsymbol{y} \\ \delta' \boldsymbol{v}_i &= \frac{\partial \boldsymbol{v}_i}{\partial \dot{\boldsymbol{y}}} \delta' \dot{\boldsymbol{y}} = \boldsymbol{J}_i^t \, \delta \dot{\boldsymbol{y}}, & \qquad \delta' \boldsymbol{\omega}_i &= \frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\boldsymbol{y}}} \delta' \dot{\boldsymbol{y}} = \boldsymbol{J}_i^r \, \delta' \dot{\boldsymbol{y}} \; . \end{aligned} \tag{2.97}$$

Since the coordinates $\boldsymbol{y}$ and $\dot{\boldsymbol{y}}$ are independent, the variations $\delta \boldsymbol{y}, \delta' \dot{\boldsymbol{y}}$ are independent as well. It follows that

$$\begin{aligned} \boldsymbol{c}^T \delta \boldsymbol{y} &= 0 \; \forall \boldsymbol{y} \\ \boldsymbol{c}^T \delta' \dot{\boldsymbol{y}} &= 0 \; \forall \dot{\boldsymbol{y}} \; , \end{aligned} \tag{2.98}$$

see [7, 25].

Similarly to the variations velocity and acceleration of each body can be expressed in terms of the generalized coordinates as

$$\begin{aligned}
\boldsymbol{v}_i &= \frac{\partial \boldsymbol{\rho}_i}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{\rho}_i}{\partial t} = \boldsymbol{J}_i^t \dot{\boldsymbol{y}} + \bar{\boldsymbol{v}}_i \\
\boldsymbol{a}_i &= \boldsymbol{J}_i^t \ddot{\boldsymbol{y}} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{J}_i^t\right) \dot{\boldsymbol{y}} + \frac{\mathrm{d}\bar{\boldsymbol{v}}_i}{\mathrm{d}t} = \boldsymbol{J}_i^t \ddot{\boldsymbol{y}} + \bar{\boldsymbol{a}}_i \\
\boldsymbol{\omega}_i &= \boldsymbol{J}_i^r \dot{\boldsymbol{y}} + \bar{\boldsymbol{\omega}}_i \\
\dot{\boldsymbol{\omega}}_i &= \boldsymbol{J}_i^r \ddot{\boldsymbol{y}} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{J}_i^r\right) \dot{\boldsymbol{y}} + \frac{\mathrm{d}\bar{\boldsymbol{\omega}}_i}{\mathrm{d}t} = \boldsymbol{J}_i^r \ddot{\boldsymbol{y}} + \bar{\boldsymbol{\zeta}}_i \ .
\end{aligned} \tag{2.99}$$

The terms of eq. (2.97) and eq. (2.99) can be inserted into the equations of d'Alembert's principle of eq. (2.84) or Jourdain's principle of eq.(2.85), respectively. For Jourdain's principle follows

$$\begin{aligned}
\delta' P = \quad & \sum_{i=1}^{n} \delta' \dot{\boldsymbol{y}}_i^T \left(\boldsymbol{J}_i^t\right)^T \left(m_i \left(\boldsymbol{J}_i^t \ddot{\boldsymbol{y}} + \bar{\boldsymbol{a}}_i\right) - \boldsymbol{f}_i^a\right) \\
& + \sum_{i=1}^{n} \delta' \dot{\boldsymbol{y}}_i^T \left(\boldsymbol{J}_i^r\right)^T \left({}^i\boldsymbol{I}_i \left(\boldsymbol{J}_i^r \ddot{\boldsymbol{y}} + \bar{\boldsymbol{\zeta}}_i\right) + \left(\boldsymbol{J}_i^r \dot{\boldsymbol{y}} + \bar{\boldsymbol{\omega}}_i\right)\tilde{} \, {}^i\boldsymbol{I}_i \left(\boldsymbol{J}_i^r \dot{\boldsymbol{y}} + \bar{\boldsymbol{\omega}}_i\right) - {}^i\boldsymbol{\ell}_i^a\right) = 0 \\
& \forall \delta' \dot{\boldsymbol{y}} \ .
\end{aligned} \tag{2.100}$$

Using the independent variation of eq. (2.98), the EOM follow as

$$\boldsymbol{M}\left(\boldsymbol{y}\right) \ddot{\boldsymbol{y}} = \bar{\boldsymbol{k}}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, t\right) - \bar{\boldsymbol{q}}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, t\right) \tag{2.101}$$

with $\boldsymbol{M}\left(\boldsymbol{y}\right)$ representing the generalized mass matrix, $\bar{\boldsymbol{k}}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, t\right)$ representing the generalized Coriolis and centrifugal forces and $\bar{\boldsymbol{q}}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, t\right)$ the generalized applied forces.
Since the EOM are derived with the independent variations $\delta' \dot{\boldsymbol{y}}$, the constraint equations are already considered in eq. (2.101), which results in a set of ODEs. The main advantage is the absence of algebraic equations and hence the option to use a broad choice of numerical integrators. Drawbacks are the usually highly nonlinear EOM and the preprocessing overhead necessary to calculate the equations.

2.2 Real-Time Capable Solvers for Multibody Systems

Numerical integration of the EOM can be done by a variety of numerical solvers, which have to be chosen according to the demand on accuracy, reliability or speed. Depending on the structure of the multibody system, it may be characterized by high numerical stiffness, strong nonlinearity of the kinematics or force elements with nonlinear behavior. These characteristics have a large influence whether a solver is suitable for a MBS or not.

For real-time (RT) capability, additional requirements have to be met for MBS. Generally speaking, a RT-simulation is computed in real-time when the integration time including output generation like visualization is lower than the physical time – the time the real-world system requires to perform the simulated behavior [16]. In [29] the term "real-time" is defined as "pertaining to the processing of data by a computer in connection with another process outside the computer according to time requirements imposed by the outside process". This definition states that RT systems are designed to continuously exchange data with other systems they interact with. Depending on the field of use, RT systems are classified as *hard real-time systems* and *soft real-time systems* [31]:

- *Hard real-time systems* are required to produce the simulation result within a time limit. Failure to produce the simulation result in time results in a failure of the whole system and has to be avoided. Typical hard RT-systems are safety-critical controllers, HiL-applications and embedded systems.

- *Soft real-time systems* are characterized to generally produce a simulation result within a given time limit. No failure will occur if the result is calculated at a later point, which means that rare events with high calculation effort, e.g. a rare contact scenario with many impacting bodies of a MBS, could lead to a higher than real-time computation time. Soft RT systems are often used for web applications or entertainment purposes like video games.

In the following sections several numerical solvers to simulate MBS in RT are shown. While hard real-time is aspired, in the following RT-capability is interpreted as the ability to determine the execution time of the solver in advance. Depending on the computer platform the calculations are executed on, this may not necessarily yield to a RT-capable simulation. However, knowledge of the execution time generally gives a good estimate whether the system can be deployed to a RT capable platform.

2.2.1 Differentiation Index of Multibody Systems

Lagrange's equations of first kind state the EOM as a system of DAE of differentiation index 3. The differentiation index hereby is defined as the number of analytical differentiations that have to be conducted on the algebraic equations in order to gain an explicit ODE [22]. Equation (2.96) can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{z}\right) \tag{2.102}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{h}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}\right) \tag{2.103}$$

$$\boldsymbol{0} = \boldsymbol{g}\left(\boldsymbol{x}\right) \ . \tag{2.104}$$

Differentiation of eq. (2.104) yields with multiple insertion of eq. (2.102) and eq. (2.103)

$$\text{index-2:} \qquad \mathbf{0} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \boldsymbol{f} \tag{2.105}$$

$$\text{index-1:} \qquad \mathbf{0} = \left(\frac{\partial}{\partial \boldsymbol{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \right) \boldsymbol{f} \right) \boldsymbol{f} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{f} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \boldsymbol{h} \,. \tag{2.106}$$

Replacing eq.(2.104) with eq. (2.106) yields differentiation index 1 since after one more time differentiation it is possible to state the time-derivative $\dot{\boldsymbol{\lambda}}$ as a function of $\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}$ if

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\lambda}} \tag{2.107}$$

is invertible [22]. Since $\boldsymbol{h} = \boldsymbol{M}^{-1} \boldsymbol{f}^a - \boldsymbol{G}^T \boldsymbol{\lambda}$, this requires that $\boldsymbol{G}$ has full row rank. When stating the system in the form

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \boldsymbol{F} \left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda} \right) \tag{2.108}$$

it can be solved with numerical integrators for ODE as shown in section 2.2.2. It is possible as well to calculate $\boldsymbol{\lambda}$ from eq. (2.106) since this forms a system of linear equations for $\boldsymbol{\lambda}$. After calculating $\boldsymbol{\lambda}$, these values can be re-inserted into eq. (2.103), forming a set of stiff ODE with eq. (2.102) and eq. (2.103). This system is solvable with suitable numerical integration methods from section 2.2.2.

2.2.2 ODE Solvers

Ordinary differential equations are of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f} \left(\boldsymbol{x}, t \right) \tag{2.109}$$

and allow the calculation of the time derivative of the state vector $\dot{\boldsymbol{x}}$ from the current state vector $\boldsymbol{x}$ and time t. Higher-order differential equations can be transformed into a system of first order equations. The probably most common method to perform a numerical integration of an ODE dates back to Leonard Euler [17] and is hence called the Euler method. As shown in fig. 2.11, an integration step of an arbitrary, scalar ODE f at time t_k is performed simply by choosing a time step size h and stating

$$\bar{x}_{k+1} = x_k + h \, f \left(x_k, t \right) \,. \tag{2.110}$$

The integration step can be employed for vector sized functions as well. Since the right-hand side is only dependent on values of step k, the method is also called the explicit Euler method.
As it can be seen in fig. 2.11, errors will occur between the approximated solution $\bar{x}_{k+1}$ and the analytical solution x_{k+1}. The total approximation error is composed of

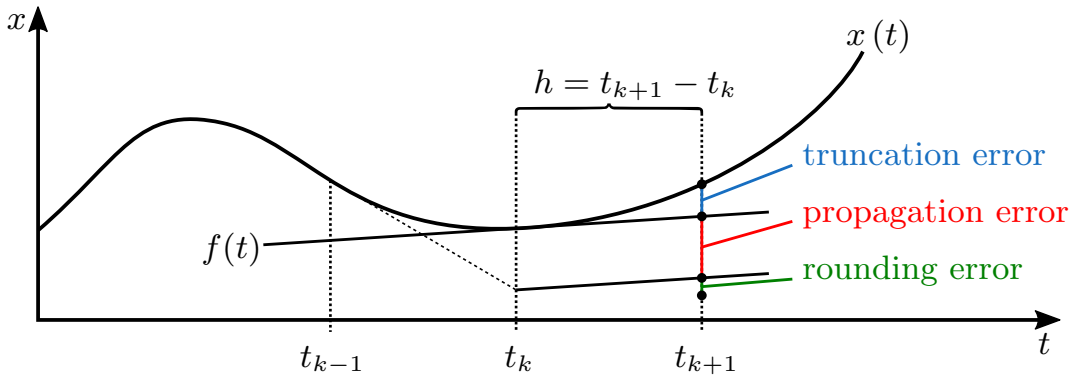


Figure 2.11: Explicit Euler integration method.

- **truncation error:** The error caused by truncation of the approximation method.
- **propagation error:** Previous integration steps are not exact, the error is carried over to the next step.
- **rounding error:** Numerical methods cause rounding errors that may become relevant for very small time steps h and long simulation times.

Choosing a smaller step size will result in smaller truncation errors but will increase numerical rounding errors depending on the data types used for numerical integration. For larger step sizes the approximation error will grow and may yield to instability of the integration. The stability of a numerical integration method, neglecting rounding errors, can be estimated for linear ODE [10]. For a linear differential equation

$$\dot{x} = \mu x \tag{2.111}$$

with complex eigenvalue μ, a stability function can be defined as

$$S(\mu) = 1 + h\mu \tag{2.112}$$

with the condition $|S(\mu)| \leq 1$, see [10, 22]. This results in a circle for the stability region for which all eigenvalues have to lie within the circle, see fig. 2.12.

For non-RT-capable methods a step size control can be implemented to ensure numerical stability and limit the approximation error. However, if the step size is not known in advance, the effort to simulate a given physical time cannot be estimated, so step size control is not suitable for RT simulation.
Besides the approximation errors the explicit Euler method has a very limited stability region. Especially for MBS with low damping, high nonlinearities and high numerical stiffness a very small step size h has to be chosen, which results in high computational effort despite the simplicity of the algorithm.
Explicit methods of higher order have been developed to circumvent the limited applicability of the explicit Euler method. Within the vast range of integration

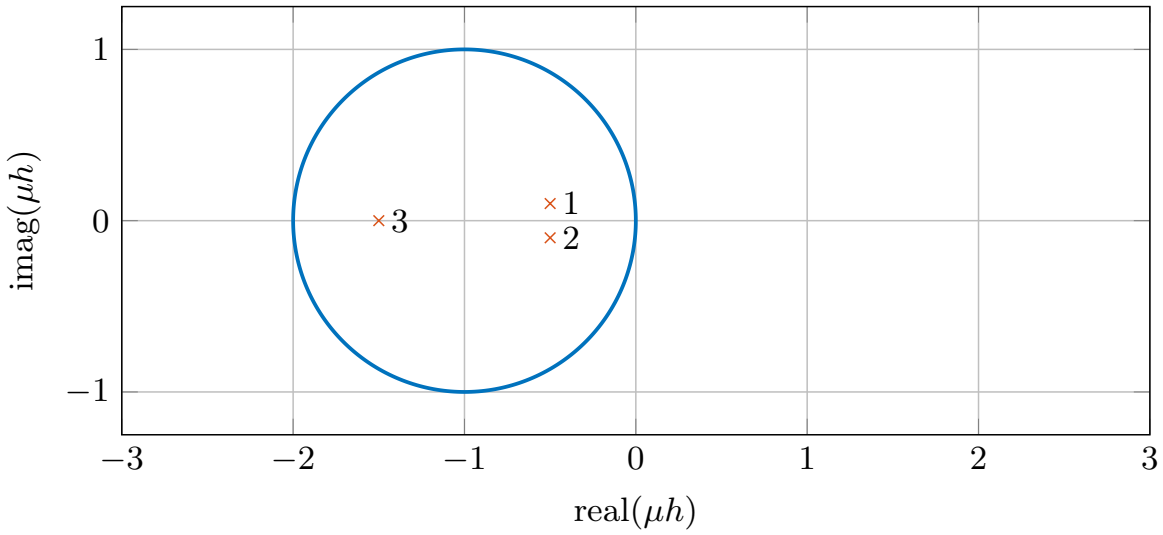


Figure 2.12: Stability region for explicit Euler with three exemplary eigenvalues.

methods the explicit Runge-Kutta (RK) methods are presented since they are of high interest for RT-capable ODE time integration. Due to its good stability properties, the 4th order RK method is shown. The order of an integration method is defined as the degree up to which the Taylor series approximation of the analytical function coincides with the integration method [22]. The 4th order RK method uses the auxiliary functions g_i (called stages) to perform one integration step. The stages may be evaluated at different positions and weighted differently for calculation of the step x_{k+1}.

$$\begin{aligned} g_1 &= f\left(x_k, t_k\right) \\ g_2 &= f\left(x_k + \frac{h}{2} g_1, t_k + \frac{h}{2}\right) \\ g_3 &= f\left(x_k + \frac{h}{2} g_2, t_k + \frac{h}{2}\right) \\ g_4 &= f\left(x_k + h\, g_3, t_k + h\right) \\ x_{k+1} &= x_k + h\left\{\frac{1}{6} g_1 + \frac{1}{3} g_2 + \frac{1}{3} g_3 + \frac{1}{6} g_4\right\} \end{aligned} \tag{2.113}$$

The stability region of the 4th order RK solver is shown in fig. 2.13 together with 1st order to 3rd order integration schemes, which is valid for all RK schemes where the order equals the number of stages. Especially for eigenvalues with low or no damping higher order RK solvers show a large region of stability and hence the step size can be chosen higher relative to the explicit Euler method, which compensates for the higher computational effort.

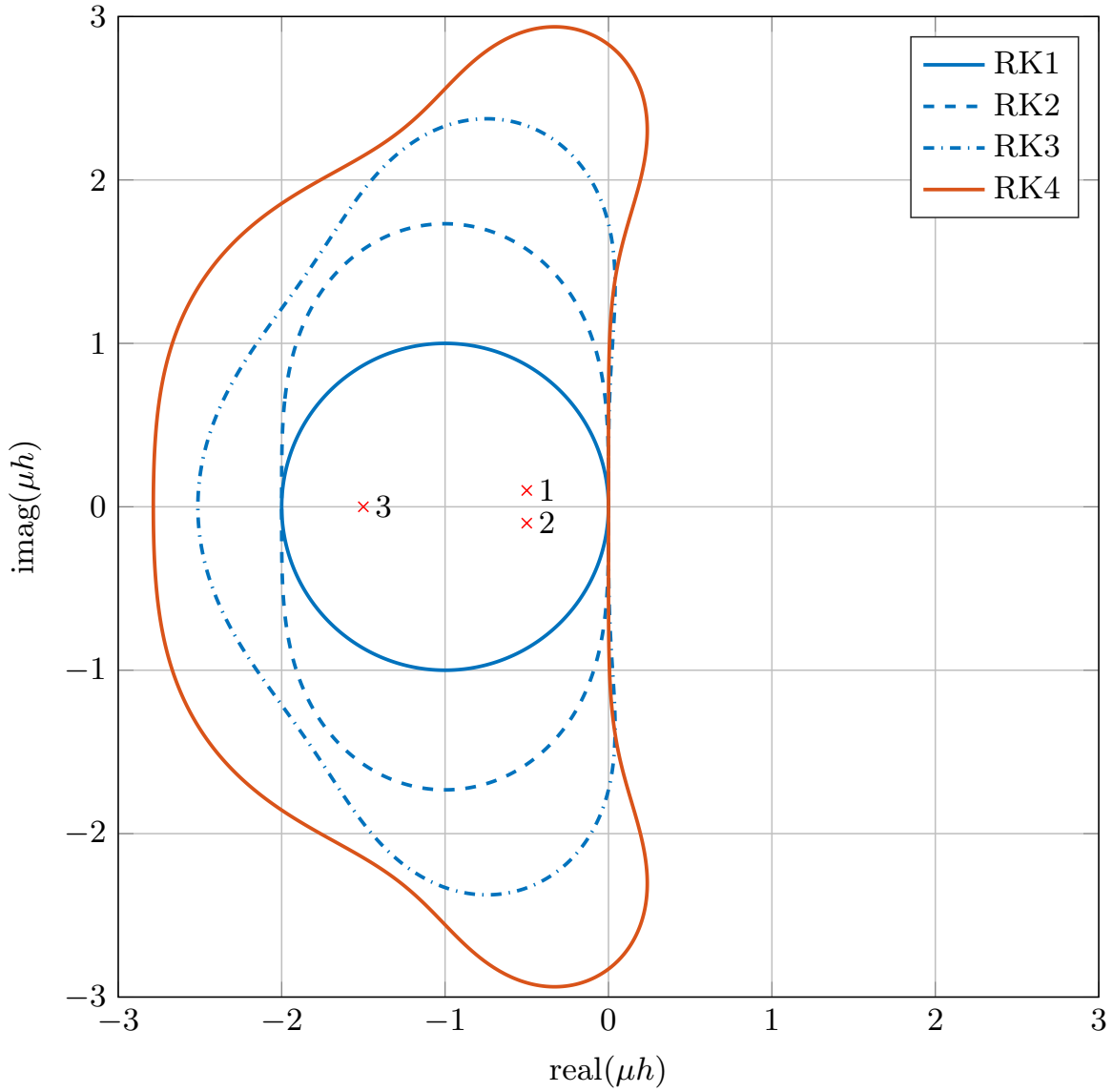


Figure 2.13: Stability regions for 1st – 4th order RK solvers.

Besides explicit integration methods implicit methods also can be employed for ODEs. For the Euler method the implicit scheme is stated as

$$x_{k+1} = x_k + h\, f\left(x_{k+1}, t\right) , \tag{2.114}$$

and it is apparent that implicit methods generally need and iterative solution of the equation (2.114) during time integration to calculate the next time step. The high calculation effort caused by the iterative processes is compensated by the large stability region, see fig. 2.14. Every eigenvalue scaled by step size h outside the circle in the right half plane is numerically stable, which in particular includes the whole left half-plane. Methods that allow numerically stable integration for all stable eigenvalues are called *A-stable* [22]. This allows integration with large step sizes even for systems with very low damping and high eigenfrequencies.

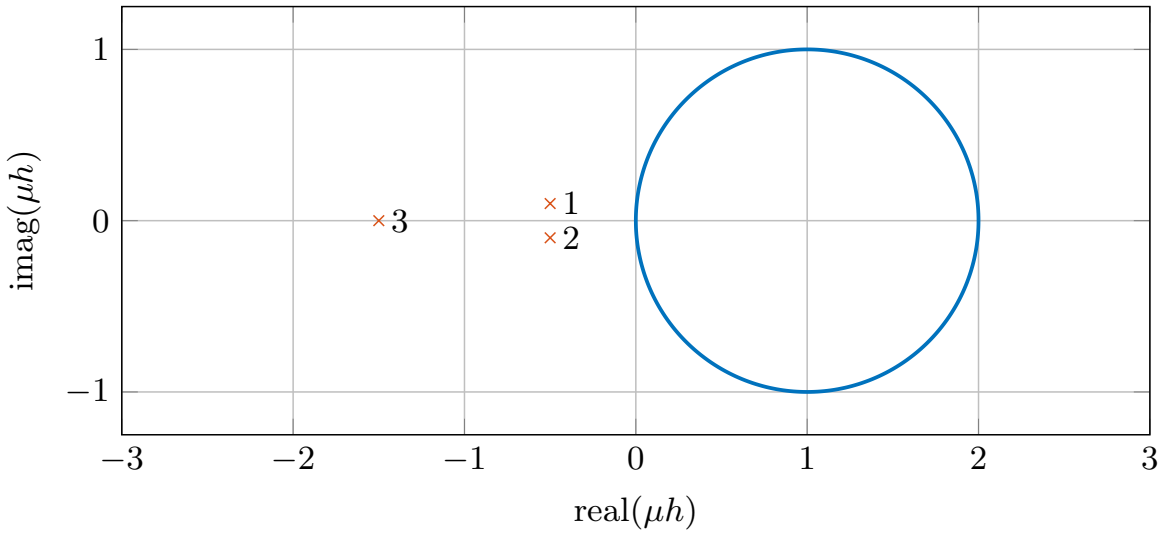


Figure 2.14: Stability region for the implicit Euler method.

The need for iterative processes is an obvious drawback for RT applications. However, it is possible to use linearly implicit methods to circumvent this problem, as it is shown in the following sections.

2.2.3 DAE Capable Solvers

Explicit methods cannot be employed for DAEs since the algebraic constraint equations will not be fulfilled without additional calculation steps. These steps have to correct the solution to lie on the manifold created by algebraic constraints, a subspace of the solutions of the differential equations. In section 2.2.1 index reduction by time differentiation is shown, which can reduce the index of a DAE to 1, resulting in a system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{z}\right) \tag{2.115}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{h}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}\right) \tag{2.116}$$

$$\boldsymbol{0} = \hat{\boldsymbol{g}}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}\right) . \tag{2.117}$$

A perturbation problem can be introduced as

$$\epsilon\dot{\boldsymbol{\lambda}} = \hat{\boldsymbol{g}}\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}\right) \tag{2.118}$$

which then can be solved with RK methods as shown in [22]. However, if the algebraic equations are stated on acceleration level, the solution has to be corrected as well. In general, solving stiff ODEs results in small step sizes and long simulation times. In [1] it is proposed to use the group of linearly implicit RK methods to solve the DAEs on index 2. These methods exist for RK solvers of all orders, but for RT applications it is useful to use first or second order methods depending on accuracy requirements. Linearly implicit integration methods

approximate $\boldsymbol{f}(\boldsymbol{x}_{k+1}, t_{k+1})$ with a first order taylor approximation as

$$\boldsymbol{f}(\boldsymbol{x}_{k+1}, t_{k+1}) \approx \boldsymbol{f}(\boldsymbol{x}_k, t_k) + h \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_k, t_k)} \dot{\boldsymbol{x}}_k \,. \tag{2.119}$$

To use linearly implicit methods, the Jacobian of the function $\boldsymbol{f}$ is required. Depending on the smoothness of the function, the Jacobian may be kept constant during run-time [2].

Linearly Implicit Euler Algorithm for Constrained MBS

Stating a DAE on index 1 and using explicit methods show high numerical effort since small integration step sizes are required. The group of linearly implicit methods allows the time integration of index 2 DAEs while still maintaining the advantages to use large integration steps and being A-stable [22].
The linearly implicit Euler method can be constructed using eq. (2.119). First, for a constrained mechanical system of index-2 one explicit integration step of eq. (2.96) can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{G}^T \\ \boldsymbol{0} & \boldsymbol{G} & \boldsymbol{0} \end{bmatrix}_k \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} h\boldsymbol{Z}\boldsymbol{z} \\ h\boldsymbol{f}^a \\ -\boldsymbol{G}\boldsymbol{z} \end{bmatrix}_k \tag{2.120}$$

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}_k + \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \end{bmatrix} . \tag{2.121}$$

Hereby the assumption was made that $\boldsymbol{G}(\boldsymbol{x}_k) \approx \boldsymbol{G}(\boldsymbol{x}_{k+1})$. This will result in an error for the algebraic equations $\boldsymbol{g}(\boldsymbol{x}_{k+1}) = \boldsymbol{0}$ and $\boldsymbol{G}(\boldsymbol{x}_{k+1})\,\boldsymbol{z}_{k+1} = \boldsymbol{0}$ which has to be corrected after the integration step.
By applying eq. (2.119) follows

$$\begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{G}^T \\ \boldsymbol{0} & \boldsymbol{G} & \boldsymbol{0} \end{bmatrix}_k \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} h\boldsymbol{Z}\boldsymbol{z} \\ h\boldsymbol{f}^a \\ -\boldsymbol{G}\boldsymbol{z} \end{bmatrix}_k + h^2 \boldsymbol{J}_k \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{2.122}$$

with the Jacobian matrix

$$\boldsymbol{J}_k = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{Z}\boldsymbol{z}) & \boldsymbol{Z} & \boldsymbol{0} \\ \frac{\partial \boldsymbol{f}^a}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{f}^a}{\partial \boldsymbol{z}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} . \tag{2.123}$$

This results in one integration step for the linearly-implicit Euler method as

$$\begin{bmatrix} \boldsymbol{E} - h\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{Z}\boldsymbol{z}) & -h\boldsymbol{Z} & \boldsymbol{0} \\ -h\frac{\partial \boldsymbol{f}^a}{\partial \boldsymbol{x}} & \boldsymbol{M} - h\frac{\partial \boldsymbol{f}^a}{\partial \boldsymbol{z}} & \boldsymbol{G}^T \\ \boldsymbol{G} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}_k \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} h\boldsymbol{Z}\boldsymbol{z} \\ h\boldsymbol{f}^a \\ -\boldsymbol{G}\boldsymbol{z} \end{bmatrix}_k \tag{2.124}$$

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}_k + \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{z} \end{bmatrix} . \tag{2.125}$$

The Jacobian $\boldsymbol{J}_k$ only covers the differential equations but not the algebraic equations. Hence this method is also called a half-explicit method [22].

2.2.4 Real-Time Capable Stabilization Methods

For index-reduced DAEs constraint drift may appear especially with small time steps or for longer simulation times due to errors occuring during the integration process, see figure 2.11. It is necessary to correct the deviation from the constraint manifold defined by $\boldsymbol{g}(\boldsymbol{x})$. In [5] a stabilization method for constraint equations is proposed by adding a correction term to the index-reduced forms eq. (2.106), eq. (2.105) as

$$\boldsymbol{G}(\boldsymbol{x})\,\boldsymbol{z} + \alpha \boldsymbol{g} = \boldsymbol{0} \tag{2.126}$$

$$\boldsymbol{G}(\boldsymbol{x})\,\dot{\boldsymbol{z}} + \frac{\mathrm{d}\boldsymbol{G}}{\mathrm{d}t}\boldsymbol{z} + 2\alpha\dot{\boldsymbol{g}} + \beta^2\boldsymbol{g} = \boldsymbol{0} \tag{2.127}$$

with the Baumgarte parameters α, β. For $\alpha, \beta > 0$ this method is asymptotically stable. However, for increasing values this method adds an artificial stiffness to the DAE and finding a good trade-off between sufficient stabilization and increasing stiffness may become cumbersome especially for real-time applications with fixed step sizes. However, easy implementation and predictable behavior still makes it a common option.
Projection methods are a suitable method to circumvent the drawbacks of Baumgarte stabilization. Instead of adding stabilization terms to the algebraic equations, the solution is projected on the constraint manifold after each integration step. This can be interpreted as a constrained minimization problem [1, 9]. For a RT capable non-iterative projection method the drift-off error $\boldsymbol{g}(\boldsymbol{x}_{k+1})$ can be eliminated with the projection

$$\Delta\boldsymbol{x} = \boldsymbol{P}\boldsymbol{g}(\boldsymbol{x}_{k+1}) \tag{2.128}$$

and the projection matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \boldsymbol{M}_k^{-1}\boldsymbol{G}_k^T\left(\boldsymbol{G}_k\boldsymbol{M}_k^{-1}\boldsymbol{G}_k^T\right)^{-1}. \tag{2.129}$$

In [9] it is shown that this projection method converges for only one projection step and hence is suitable for RT simulations. Another advantage of this method is the option to use the already known matrices $\boldsymbol{G}_k, \boldsymbol{M}_k$.

2.3 Flexible Multibody Systems

Flexible multibody systems are systems that allow the deformations of continua within the rigid body motion of elastic or rigid bodies. According to [4], multibody systems can be classified as rigid body systems, linearly elastic multibody systems and nonlinearly elastic multibody systems. Linearly elastic multibody

systems combine the large rigid body motion with small, linearizable elastic deformation. These systems can be modeled with a body-related floating frame of reference (FFoR) that is used to separate the rigid body motion from the elastic deformation [67]. Nonlinearly elastic multibody systems can be modeled for example with the absolute nodal coordinate formulation, see [26, 59]. Since this work covers modeling and simulation of road vehicles with small elastic deformations, the FFoR approach is used to govern the EOM of the MBS.

The elastic deformation is assumed to be small and hence it is linearized with respect to a set of elastic coordinates $\boldsymbol{q}_i$. While section 2.1 describes the derivation of EOM for rigid multibody systems, this section covers the kinematic and kinetic analysis of flexible multibody systems. Therefore the Newton and Euler equations are extended by flexible DOF and all necessary kinematic and kinetic properties are re-evaluated for flexible bodies.

First, the FFoR approach is presented. This approach is known from [57, 59] and makes use of the fact that when describing the body in a body-related, moving reference frame, the mass matrix is only dependent on the elastic coordinates. Afterwards, a mixed approach is shown which describes the translational motion in the global reference frame but the rotational motion remains expressed in the moving reference system.

The EOM for flexible multibody systems are derived similar to the EOM for rigid bodies by using Jourdain's principle. They are complemented with the power generated by the deformation of the continua. The theory of linear elasticity is used to govern the properties of elastic bodies. A comprehensive overview on continuum mechanics in multibody systems and the derivation of elastic body properties can be found in [57, 58, 59].

The deformation behavior of an elastic body is required to represent an approximation of a real continuum. Based on homogeneous, isotropic material theory, the elastic behavior can be derived from beam theory or using the finite element method. The elastic properties are collected in a structure called standard input data (SID) [57, 67], which allows the storage of all matrices necessary to evaluate elastic bodies and an approximation of the matrices as taylor series. Whenever possible, the equations derived in this work makes use of the SID to simplify the numerical evaluation.

2.3.1 The Floating Frame of Reference Approach

To simplify evaluation of the EOM, it is useful to separate the rigid body motion and elastic deformation and to describe the EOM in moving reference frames that are related to the bodies of the multibody system. Figure 2.15 shows an elastic body in a reference state Ω_0 and deformed state Ω_1. The moving reference frame and the position of a point k change during the movement and deformation of a body.

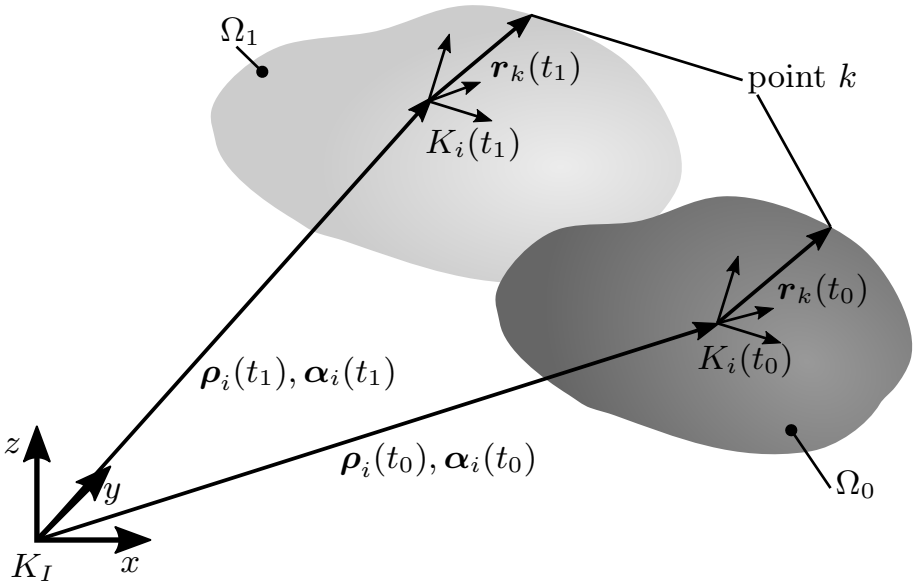


Figure 2.15: Floating frame of reference of a body i.

A point k on body i is described with the position and orientation of a frame fixed to point k given as

$$ {}^{i}\boldsymbol{\rho}_k = {}^{i}\boldsymbol{\rho}_i + {}^{i}\boldsymbol{r}_k = {}^{i}\boldsymbol{\rho}_i + {}^{i}\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i \tag{2.130} $$

with the position vector of point k relative to the reference frame divided into a constant part $\boldsymbol{R}_k$ in undeformed configuration and the elastic deformation

$$ \boldsymbol{u}_{i,k} = \boldsymbol{\Phi}_k \boldsymbol{q}_i \,. \tag{2.131} $$

Since a linear Ritz approach is used, the time- and location-dependent deformation vector $\boldsymbol{u}_{i,k}$ is calculated with a $3 \times n_q$ matrix $\boldsymbol{\Phi}_k$ only dependent on the position of the point k and the $n_q \times 1$ vector of time-dependent elastic coordinates $\boldsymbol{q}_i$.
Respectively, the current orientation of a coordinate system attached to point k relative to the global frame, $\boldsymbol{A}_k$, is also dependent on the elastic coordinates with

$$ \boldsymbol{A}_k = \boldsymbol{B}_k \boldsymbol{A}_i = \left(\boldsymbol{E} - \widetilde{\boldsymbol{\Psi}_k \boldsymbol{q}_i} \right) \boldsymbol{\Gamma}_k \boldsymbol{A}_i \,. \tag{2.132} $$

Here, small deformation angles

$$ \boldsymbol{\vartheta}_{i,k} = \boldsymbol{\Psi}_k \boldsymbol{q}_i \tag{2.133} $$

are approximated similar to the translational deformation. For small angles the rotation matrix of eq. (2.13) can be linearized to $\boldsymbol{B}_k = (\boldsymbol{E} - \widetilde{\boldsymbol{\vartheta}}_k)\boldsymbol{\Gamma}_k$ with the constant rotation matrix $\boldsymbol{\Gamma}_k$ defining an initial offset.
Translational coordinates $\boldsymbol{\rho}_i$ and rotation parametrization $\boldsymbol{\alpha}_i$ for the FFoR together with the elastic coordinates $\boldsymbol{q}_i$ form the position vector $\boldsymbol{x}_i$ of body i

with

$$\boldsymbol{x}_i = \begin{bmatrix} {}^i\boldsymbol{\rho}_i \\ \boldsymbol{\alpha}_i \\ \boldsymbol{q}_i \end{bmatrix} . \tag{2.134}$$

These body coordinates are all expressed in the moving reference frame and are related to the state-space formulation of eq. (2.69). For calculating the velocity quantities similar to eq. (2.70), relative motion of the reference frame has to be taken into account. For the derivative with respect to the inertial system of the velocity follows

$${}^i\boldsymbol{v}_i = {}^i\dot{\boldsymbol{\rho}}_i + {}^i\widetilde{\boldsymbol{\omega}}_i\,{}^i\boldsymbol{\rho}_i, \quad {}^i\dot{\boldsymbol{\rho}}_i = {}^i\boldsymbol{v}_i + {}^i\widetilde{\boldsymbol{\rho}}_i\,{}^i\boldsymbol{\omega}_i \tag{2.135}$$

and the velocity vector $\boldsymbol{z}_i$ is then

$$\boldsymbol{z}_i = \begin{bmatrix} {}^i\boldsymbol{v}_i \\ {}^i\boldsymbol{\omega}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix} \text{ with } \dot{\boldsymbol{x}}_i \neq \boldsymbol{z}_i\,. \tag{2.136}$$

Again, the vectors $\boldsymbol{z}_i$ and $\dot{\boldsymbol{x}}_i$ are connected with the kinematic matrix $\boldsymbol{Z}_i$. For the FFoR-approach this matrix is

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{Z}_i\boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{E} & {}^i\widetilde{\boldsymbol{\rho}}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_r(\boldsymbol{\alpha}_i) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \boldsymbol{z}_i \tag{2.137}$$

with $\boldsymbol{Z}_r(\boldsymbol{\alpha}_i)$ given in eq. (2.61) for Tait-Bryan-angles and in eq. (2.31) for quaternions.
For the translational and angular velocity of a point k on body i in coordinates of the FFoR then follows from eq. (2.130)

$$\begin{aligned} {}^i\boldsymbol{v}_k &= {}^i\boldsymbol{v}_i + \boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i + {}^i\widetilde{\boldsymbol{\omega}}_i\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k\boldsymbol{q}_i\right) \\ &= \begin{bmatrix} \boldsymbol{E} & -\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k\boldsymbol{q}_i\right)\widetilde{\ } & \boldsymbol{\Phi}_k \end{bmatrix} \boldsymbol{z}_i \\ &= \boldsymbol{T}_k^t\,\boldsymbol{z}_i \end{aligned} \tag{2.138}$$

and for the angular velocity

$$\begin{aligned} {}^i\boldsymbol{\omega}_k &= {}^i\boldsymbol{\omega}_i + \boldsymbol{\Psi}_k\boldsymbol{q}_i \\ &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{E} & \boldsymbol{\Psi}_k \end{bmatrix} \boldsymbol{z}_i \\ &= \boldsymbol{T}_k^r\,\boldsymbol{z}_i \end{aligned} \tag{2.139}$$

Using the matrices $\boldsymbol{T}_k^t$ and $\boldsymbol{T}_k^r$ simplifies the derivation of further expressions within multibody systems, especially for constraint equations and discrete forces and moments.
The acceleration of a point k follows by a second derivation from eq. (2.138), eq. (2.139) as

$${}^i\boldsymbol{a}_k = \boldsymbol{T}_k^t\,\dot{\boldsymbol{z}}_i + {}^i\widetilde{\boldsymbol{\omega}}_i\left({}^i\boldsymbol{v}_i + 2\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i + {}^i\widetilde{\boldsymbol{\omega}}_i\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k\boldsymbol{q}_i\right)\right)$$

$$= \boldsymbol{T}_k^t \dot{\boldsymbol{z}}_i + \boldsymbol{\zeta}_k^t \tag{2.140}$$

and for the angular acceleration

$$\begin{aligned} {}^i\boldsymbol{b}_k &= \boldsymbol{T}_k^r \dot{\boldsymbol{z}}_i + \boldsymbol{\Psi}_k \dot{\boldsymbol{q}}_i \\ &= \boldsymbol{T}_k^r \dot{\boldsymbol{z}}_i + \boldsymbol{\zeta}_k^r \end{aligned} \tag{2.141}$$

The kinematic properties of the FFoR approach can be used to describe every point on a continuum. The description of a point k only relies on the vector $\boldsymbol{R}_k$ and the matrices $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$, $\boldsymbol{\Gamma}_k$. These quantities are stored in the SID structure mentioned above.

2.3.2 Mixed Reference Frame Approach

While the FFoR approach describes all variables in the floating frame K_i, this mixed approach uses the global system K_I for the position coordinate $\boldsymbol{\rho}_i$ [13, 16]. However, the points of body i are still described in the floating frame K_i. This yields to the description of a point k in the FFoR as

$${}^i\boldsymbol{\rho}_k = \boldsymbol{A}_i {}^I\boldsymbol{\rho}_i + {}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i \tag{2.142}$$

$$\boldsymbol{A}_k = \left(\boldsymbol{E} - \widetilde{\boldsymbol{\Psi}_k \boldsymbol{q}_i}\right) \boldsymbol{\Gamma}_k \boldsymbol{A}_i \tag{2.143}$$

The velocity and acceleration of the point k are calculated as

$$\begin{aligned} {}^i\boldsymbol{v}_k &= \boldsymbol{A}_i {}^I\boldsymbol{v}_i + \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + {}^i\widetilde{\boldsymbol{\omega}}_i \left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right) \\ &= \begin{bmatrix} \boldsymbol{A}_i & -\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\widetilde{\ } & \boldsymbol{\Phi}_k \end{bmatrix} \boldsymbol{z}_i \\ &= \boldsymbol{T}_k^t \boldsymbol{z}_i \end{aligned} \tag{2.144}$$

$$\begin{aligned} {}^i\boldsymbol{a}_k &= \boldsymbol{A}_i {}^I\boldsymbol{a}_i + \boldsymbol{\Phi}_k \ddot{\boldsymbol{q}}_i + 2{}^i\widetilde{\boldsymbol{\omega}}_i \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + \left({}^i\dot{\widetilde{\boldsymbol{\omega}}}_i + {}^i\widetilde{\boldsymbol{\omega}}{}^i\widetilde{\boldsymbol{\omega}}\right) \left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right) \\ &= \boldsymbol{T}_k^t \dot{\boldsymbol{z}}_i + \boldsymbol{\zeta}_k^t \end{aligned} \tag{2.145}$$

with $\boldsymbol{x}_i$, $\boldsymbol{z}_i$ now containing the states of the body i as

$$\boldsymbol{x}_i = \begin{bmatrix} {}^I\boldsymbol{\rho}_i \\ \boldsymbol{\alpha}_i \\ \boldsymbol{q}_i \end{bmatrix}, \; \boldsymbol{z}_i = \begin{bmatrix} {}^I\boldsymbol{v}_i \\ {}^i\boldsymbol{\omega}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix}. \tag{2.146}$$

Since ${}^I\boldsymbol{\rho}_i$, ${}^I\boldsymbol{v}_i$ and ${}^I\boldsymbol{a}_i$ are described in the global reference frame, it is apparent that

$${}^I\boldsymbol{v}_i = \frac{\mathrm{d}{}^I\boldsymbol{\rho}_i}{\mathrm{d}t}, \; {}^I\boldsymbol{a}_i = \frac{\mathrm{d}{}^I\boldsymbol{v}_i}{\mathrm{d}t} \tag{2.147}$$

Since the translational entries are expressed with respect to the global coordinate system, the mixed reference frame approach results in a kinematic matrix in block-diagonal form

$$\dot{\boldsymbol{x}}_i = \boldsymbol{Z}_i \boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_r(\boldsymbol{\alpha}_i) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \boldsymbol{z}_i \tag{2.148}$$

While this approach is quite similar to the standard floating frame of reference approach, stating the translational components in the global reference frame yields to less rapidly changing position values especially for fast rotating bodies. This simplifies the Jacobian calculation which is necessary for implicit and semi-implicit solvers and may allow to keep the Jacobian constant during the simulation. For example, figure 2.16 shows a wheel with radius $r = 0.3\,\mathrm{m}$ rolling slip-free on an even surface with constant speed $v_x = 3\,\frac{m}{s}$. The Jacobian $\frac{\partial}{\partial \boldsymbol{z}_i}(\boldsymbol{Z}_i \boldsymbol{z}_i)$ of eq. (2.93) of both the FFoR and the mixed reference frame are compared and show that $\boldsymbol{Z}_i$ is independent of $\boldsymbol{\rho}_i$ for the mixed reference frame while the wheel position ${}^i\boldsymbol{\rho}_i$ in local coordinates is required for the FFoR kinematic matrix, see fig. 2.16.

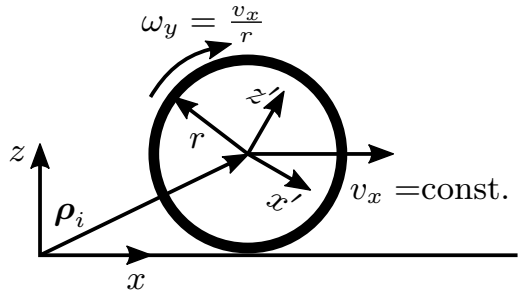


Figure 2.16: Pure kinematic rolling.

FFoR:

$$\frac{\partial}{\partial \boldsymbol{z}_i}(\boldsymbol{Z}_i \boldsymbol{z}_i) = \boldsymbol{Z}_i = \begin{bmatrix} \boldsymbol{E} & {}^i\widetilde{\boldsymbol{\rho}}_i \\ \boldsymbol{0} & \frac{1}{2}\boldsymbol{L}_i^T \end{bmatrix}$$

Mixed reference frame:

$$\frac{\partial}{\partial \boldsymbol{z}_i}(\boldsymbol{Z}_i \boldsymbol{z}_i) = \boldsymbol{Z}_i = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{2}\boldsymbol{L}_i^T \end{bmatrix}$$

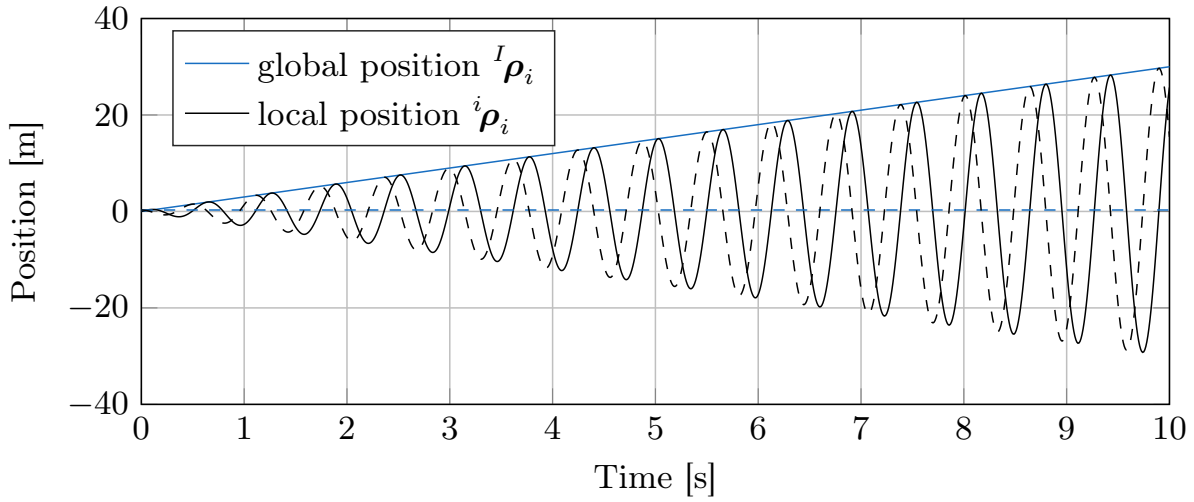


Figure 2.17: Global and local position of wheel (solid: x-pos., dashed: z-pos.).

The effect of ${}^i\boldsymbol{\rho}_i$ on the Jacobian of the FFoR approach is pointed out when comparing the position in the global and local reference frame, see figure 2.17. The entries of ${}^i\boldsymbol{\rho}_i$ will change with increasing velocity for increasing distance from

the origin, hence creating rapidly changing parameters within the Jacobian. This increases the truncation error especially for low order semi-implicit methods. Similar effects can be shown for the Jacobian of applied forces.

2.3.3 Kinetics in Flexible Multibody Systems

The EOM of flexible multibody systems can be evaluated with the same principles as rigid multibody systems. While in eq. (2.84) and eq. (2.85) internal forces do not occur, the virtual work and virtual power of internal forces have to be considered for flexible bodies. For deformable continua D'Alembert's principle and Jourdain's principle are stated in integral form preferably. The relative orientation of the points on the continua changes due to deformation and hence inertial properties are not constant. A detailed derivation of the theory of deformable bodies and their application is given in [4, 57, 59]. Based on these references the necessary equations to gain the elastic properties for flexible multibody systems are stated in the following. Jourdain's principle for a flexible body is given as

$$\delta P_i = \delta P_i^{\text{inertial}} + \delta P_i^{\text{internal}} - \delta P_i^{\text{body}} - \delta P_i^{\text{surface}} - \delta P_i^{\text{discrete}} \,. \tag{2.149}$$

For linear-elastic materials, the virtual power of internal forces can be calculated from the stress and strain tensor, which can be rearranged in vector form as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{13} & \sigma_{23} \end{bmatrix}^T \tag{2.150}$$

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & \epsilon_{33} & \epsilon_{12} & \epsilon_{13} & \epsilon_{23} \end{bmatrix}^T \,, \tag{2.151}$$

and generalized Hooke's law is stated as

$$\boldsymbol{\sigma} = \hat{\boldsymbol{E}}\boldsymbol{\epsilon} \,. \tag{2.152}$$

Using the time derivative $\dot{\boldsymbol{\epsilon}}$ of the strain vector the virtual power of a continuum Ω_0 in reference configuration is

$$\delta P_i^{\text{internal}} = \int_{\Omega_0} \boldsymbol{\sigma}^T \delta' \dot{\boldsymbol{\epsilon}} \mathrm{d}V \,. \tag{2.153}$$

For small strains, the strain vector $\boldsymbol{\epsilon}$ can be calculated from the displacement vector $\boldsymbol{u}$ and the deformation tensor $\boldsymbol{D}$ with the linear relation

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{23} \end{bmatrix} = \boldsymbol{D}\boldsymbol{u} \text{ , with } \quad \boldsymbol{D} = \frac{1}{2} \begin{bmatrix} 2\frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & 2\frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & 2\frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \end{bmatrix} \tag{2.154}$$

where x_j are the coordinates of the material points of the undeformed continuum. Using the Ritz-ansatz from eq. (2.131) results in

$$\boldsymbol{\epsilon} = \boldsymbol{D}\boldsymbol{\Phi}\boldsymbol{q}\,. \tag{2.155}$$

Inserted in Hooke's law gives

$$\boldsymbol{\sigma} = \hat{\boldsymbol{E}}\boldsymbol{D}\boldsymbol{\Phi}\boldsymbol{q}\,. \tag{2.156}$$

for the stresses occurring at each material point. The tensor $\hat{\boldsymbol{E}}$ is the matrix of elastic coefficients [60]. For homogeneous isotropic materials two constants, known as the Lamé constants [60], are defined as

$$\lambda = \frac{\gamma E}{(1+\gamma)\,(1-2\gamma)} \tag{2.157}$$

$$\mu = \frac{E}{2\,(1+\gamma)} \tag{2.158}$$

with Young's modulus E and Poisson ratio γ. The matrix of elastic coefficients is then defined as

$$\hat{\boldsymbol{E}} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{bmatrix} \tag{2.159}$$

With all elastic properties defined with respect to the elastic coordinates, geometrical properties and material laws, eq. (2.153) as the virtual power of the inner forces can be written as

$$\delta P_{\mathrm{i}} = \int_{\Omega_0} \delta\left(\boldsymbol{D}\boldsymbol{\Phi}\dot{\boldsymbol{q}}\right)^T \hat{\boldsymbol{E}}\boldsymbol{D}\boldsymbol{\Phi}\boldsymbol{q}\,\mathrm{d}V = \int_{\Omega_0} \delta'\dot{\boldsymbol{q}}^T \boldsymbol{D}^T\boldsymbol{\Phi}^T\hat{\boldsymbol{E}}\boldsymbol{D}\boldsymbol{\Phi}\boldsymbol{q}\,\mathrm{d}V\,. \tag{2.160}$$

Thus, for flexible multibody systems, Jourdain's principle extends to [57]

$$\begin{aligned} \delta P_i = & \underbrace{\int_{\Omega_{0,i}} \delta'\boldsymbol{v}^T\boldsymbol{a}\rho_0\mathrm{d}V}_{\text{inertial forces}} + \underbrace{\int_{\Omega_{0,i}} \begin{bmatrix}\boldsymbol{0} \\ \delta'\dot{\boldsymbol{q}}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{D}^T\boldsymbol{\Phi}^T\hat{\boldsymbol{E}}\boldsymbol{D}\boldsymbol{\Phi}\end{bmatrix}\begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{q}\end{bmatrix}\mathrm{d}V}_{\text{internal forces}} \\ & - \underbrace{\int_{\Omega_{0,i}} \delta'\boldsymbol{v}^T\boldsymbol{b}_0\rho_0\mathrm{d}V}_{\text{body forces}} - \underbrace{\int_{\Omega_{0,i}} \delta'\boldsymbol{v}^T\boldsymbol{p}_0\mathrm{d}A}_{\text{surface traction}} \\ & - \underbrace{\sum_k \delta'\boldsymbol{v}_{i,k}^T\boldsymbol{f}_{i,k}}_{\text{discrete forces}} - \underbrace{\sum_k \delta'\boldsymbol{\omega}_{i,k}^T\boldsymbol{\ell}_{i,k}^a}_{\text{discrete moments}}\,. \end{aligned} \tag{2.161}$$

In order to gain the EOM, it is necessary to evaluate the integrals in eq. (2.161). The evaluation will result in different EOM for the FFoR approach and the mixed reference approach. To evaluate the integrals, the variation of eq. (2.138) and eq. (2.144) is used with

$$\delta' \boldsymbol{v}_k = \boldsymbol{T}_k^t \, \delta' \boldsymbol{z}_i \tag{2.162}$$

and inserted in eq. (2.161) with eq. (2.140) and eq. (2.145) results in four integrals to evaluate:

$$\int_{\Omega_{0,i}} \delta' \boldsymbol{z}_i^T \boldsymbol{T}_k^{t\,T} \boldsymbol{a}_k \rho_0 \mathrm{d}V = \delta' \boldsymbol{z}_i^T \int_{\Omega_{0,i}} \boldsymbol{T}_k^{t\,T} \boldsymbol{T}_k^t \rho_0 \mathrm{d}V \ddot{\boldsymbol{z}}_i + \delta' \boldsymbol{z}_i^T \int_{\Omega_{0,i}} \boldsymbol{T}_k^{t\,T} \boldsymbol{\zeta}_k^t \rho_0 \mathrm{d}V \tag{2.163}$$

$$\int_{\Omega_{0,i}} \begin{bmatrix} \boldsymbol{0} \\ \delta' \dot{\boldsymbol{q}}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D}^T \boldsymbol{\Phi}^T \hat{\boldsymbol{E}} \boldsymbol{D} \boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q}_i \end{bmatrix} \mathrm{d}V = \delta' \boldsymbol{z}_i^T \begin{bmatrix} \boldsymbol{0} \\ \int_{\Omega_{0,i}} \boldsymbol{D}^T \boldsymbol{\Phi}^T \hat{\boldsymbol{E}} \boldsymbol{D} \boldsymbol{\Phi} \, \mathrm{d}V \boldsymbol{q}_i \end{bmatrix} \tag{2.164}$$

$$\int_{\Omega_{0,i}} \delta' \boldsymbol{v}^T \boldsymbol{b}_0 \rho_0 \mathrm{d}V = \delta' \boldsymbol{z}_i^T \int_{\Omega_{0,i}} \boldsymbol{T}_k^{t\,T} \boldsymbol{b}_0 \rho_0 \mathrm{d}V \tag{2.165}$$

$$\int_{\Omega_{0,i}} \delta' \boldsymbol{v}^T \boldsymbol{p}_0 \mathrm{d}A = \delta' \boldsymbol{z}_i^T \int_{\Omega_{0,i}} \boldsymbol{T}_k^{t\,T} \boldsymbol{p}_0 \mathrm{d}A \tag{2.166}$$

Evaluation of eq. (2.163) – (2.166) requires knowledge of the mechanical properties of the continua modeled within the FMBS. For rigid bodies, the matrix $\boldsymbol{T}_k^t$ is constant and the evaluation of the integrals is quite simple, resulting in constant inertial properties. The integrals of flexible bodies on the other hand are dependent on the flexible coordinates of the body and require a recalculation during the simulation of a FMBS. In [57, 60, 67] the integrals are evaluated as a taylor approximation of the flexible states, which circumvents the evaluation of the integrals during the simulation with a pre-processing step. The required data is stored in a standardized structure called SID, which can be calculated from FE-data, beam theory or for simple structures from analytical solutions.

2.3.4 SID Evaluation for the Mixed Reference Frame Approach

Since the $\boldsymbol{T}_k^t$-matrix of eq. (2.144) is different from the FFoR approach, eq. (2.164), eq. (2.165) and eq. (2.166) have to be reevaluated for the mixed reference frame approach of section 2.3.2. Thereby it is ensured that the resulting terms can be calculated using existing SID structures by converting the integrals of the mixed reference frame approach to the same integrals that are evaluated for the FFoR approach in [67].

First, the inertial forces of eq. (2.163) are re-evaluated. With eq. (2.144) follows for the mass matrix

$$\begin{aligned}\boldsymbol{M}_i &= \int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T \\ \left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \\ \boldsymbol{\Phi}_k^T \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_i & -\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} & \boldsymbol{\Phi}_k \end{bmatrix} \mathrm{d}m \\ &= \begin{bmatrix} \boldsymbol{M}_{tt} & & \text{sym.} \\ \boldsymbol{M}_{rt} & \boldsymbol{M}_{rr} & \\ \boldsymbol{M}_{et} & \boldsymbol{M}_{er} & \boldsymbol{M}_{ee} \end{bmatrix} \end{aligned} \tag{2.167}$$

and for the Coriolis and centrifugal forces

$$\boldsymbol{h}_\omega = -\int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T \\ \left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \\ \boldsymbol{\Phi}_k^T \end{bmatrix} \left[\widetilde{\boldsymbol{\omega}}_i \left(2\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{\omega}}_i \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\right)\right] \mathrm{d}m\,. \tag{2.168}$$

Mass Matrix Evaluation

Piecewise comparison of the terms of eq. (2.167) will result in six different integrals to be evaluated.
The first term represents the translational rigid body motion and is equal to the mass of the body:

$$\boldsymbol{M}_{tt} = \int_{\Omega_0} \boldsymbol{A}_i^T \boldsymbol{A}_i \mathrm{d}m = m_i \boldsymbol{E} \tag{2.169}$$

It is of course unchanged compared to the standard FFoR approach. The second term yields

$$\boldsymbol{M}_{rt} = \int_{\Omega_0} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \boldsymbol{A}_i \mathrm{d}m = \int_{\Omega_0} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \mathrm{d}m \boldsymbol{A}_i = m_i \widetilde{\boldsymbol{c}}_i \boldsymbol{A}_i \tag{2.170}$$

and represents the location of the CoG relative to the reference frame.
The inertia tensor for elastic bodies is calculated as

$$\boldsymbol{M}_{rr} = -\int_{\Omega_0} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \mathrm{d}m = \boldsymbol{I}_i \tag{2.171}$$

The fourth and fifth term couple the motion of the reference system to the elastic coordinates:

$$\boldsymbol{M}_{et} = \int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{A}_i \mathrm{d}m = \int_{\Omega_0} \boldsymbol{\Phi}_k^T \mathrm{d}m \boldsymbol{A}_i = \boldsymbol{C}_i^t \boldsymbol{A}_i \tag{2.172}$$

$$\boldsymbol{M}_{er} = \int_{\Omega_0} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)^{\sim} \mathrm{d}m = \boldsymbol{C}_i^r \tag{2.173}$$

The elastic mass matrix, calculated from the sixth term, is

$$\boldsymbol{M}_{ee} = \int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k \mathrm{d}m = \boldsymbol{M}_i^e. \tag{2.174}$$

Using eq. (2.169) – eq. (2.174) the mass matrix for elastic bodies formulated in the mixed reference frame approach results in

$$\boldsymbol{M}_i = \begin{bmatrix} m_i \boldsymbol{E} & & \text{sym.} \\ m_i \widetilde{\boldsymbol{c}}_i \boldsymbol{A}_i & \boldsymbol{I}_i & \\ \boldsymbol{C}_i^t \boldsymbol{A}_i & \boldsymbol{C}_i^r & \boldsymbol{M}_i^e \end{bmatrix} \tag{2.175}$$

Coriolis and Centrifugal Forces

The Coriolis and centrifugal forces of eq. (2.168) can be evaluated piecewise similar to the mass matrix.
The forces for translational motion are

$$\begin{aligned} \boldsymbol{h}_{\omega t} &= -\int_{\Omega_0} \boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \left(2\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{\omega}}_i \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\right) \mathrm{d}m \\ &= -2\boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \int_{\Omega_0} \boldsymbol{\Phi}_k \mathrm{d}m \dot{\boldsymbol{q}}_i - \boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \widetilde{\boldsymbol{\omega}}_i \int_{\Omega_0} \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right) \mathrm{d}m \\ &= -2\boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \boldsymbol{C}_i^{tT} - \boldsymbol{A}_i^T m_i \widetilde{\boldsymbol{\omega}}_i \widetilde{\boldsymbol{\omega}}_i \boldsymbol{c}_i \,. \end{aligned} \tag{2.176}$$

The forces for rotational motion are

$$\begin{aligned} \boldsymbol{h}_{\omega r} &= -\int_{\Omega_0} \widetilde{\left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)} \widetilde{\boldsymbol{\omega}}_i \left(2\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + \widetilde{\boldsymbol{\omega}}_i \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\right) \mathrm{d}m \\ &= -\int_{\Omega_0} 2 \widetilde{\left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)} \widetilde{\left(\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i\right)} \mathrm{d}m \, \boldsymbol{\omega}_i \\ &\quad - \widetilde{\boldsymbol{\omega}}_i \int_{\Omega_0} \widetilde{\left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)} \widetilde{\left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)} \mathrm{d}m \, \boldsymbol{\omega}_i \\ &= -\sum_{l=1}^{n_q} \boldsymbol{G}_l^r \dot{q}_l \, \boldsymbol{\omega}_i - \widetilde{\boldsymbol{\omega}}_i \boldsymbol{I}_i \boldsymbol{\omega}_i \end{aligned} \tag{2.177}$$

with $\boldsymbol{G}^r$ a $3 \times n_q \times 3$-matrix representing the forces on the rotational motion caused by the velocity of the elastic coordinates.

The forces on the elastic coordinates are

$$\begin{aligned}\boldsymbol{h}_{\omega e} &= -\int_{\Omega_0} \boldsymbol{\Phi}_k^T \widetilde{\boldsymbol{\omega}}_i \left(2\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i - \widetilde{\boldsymbol{\omega}}_i \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\right) \mathrm{d}m \\ &= -\int_{\Omega_0} 2\boldsymbol{\Phi}_k^T \left(2\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i\right)\widetilde{\ } \mathrm{d}m\, \boldsymbol{\omega}_i - \boldsymbol{\omega}_i^T \int_{\Omega_0} \widetilde{\boldsymbol{\Phi}}_k^T \left(\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)\widetilde{\ } \mathrm{d}m\, \boldsymbol{\omega}_i \\ &= -\sum_{l=1}^{n_q} \boldsymbol{G}_l^e \dot{q}_l\, \boldsymbol{\omega}_i - \boldsymbol{O}_i^e \boldsymbol{\omega}_q \end{aligned} \tag{2.178}$$

with $\boldsymbol{G}^e$ a $n_q \times n_q \times 3$-matrix representing the forces on elastic coordinates caused by the velocity of the elastic coordinates. $\boldsymbol{\omega}_q$ is a 6×1-vector with $\boldsymbol{\omega}_q = \begin{bmatrix}\omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_1\omega_2 & \omega_2\omega_3 & \omega_1\omega_3\end{bmatrix}$, see [67].
Summarized the Coriolis and centrifugal forces for the mixed reference frame approach are

$$\boldsymbol{h}_\omega = \begin{bmatrix}\boldsymbol{h}_{\omega t} \\ \boldsymbol{h}_{\omega r} \\ \boldsymbol{h}_{\omega e}\end{bmatrix} = \begin{bmatrix} -2\boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \boldsymbol{C}_i^{tT} - \boldsymbol{A}_i^T m_i \widetilde{\boldsymbol{\omega}}_i \widetilde{\boldsymbol{\omega}}_i \boldsymbol{c}_i \\ -\sum_{l=1}^{n_q} \boldsymbol{G}_l^r \dot{q}_l\, \boldsymbol{\omega}_i - \widetilde{\boldsymbol{\omega}}_i \boldsymbol{I}_i \boldsymbol{\omega}_i \\ -\sum_{l=1}^{n_q} \boldsymbol{G}_l^e \dot{q}_l\, \boldsymbol{\omega}_i - \boldsymbol{O}_i^e \boldsymbol{\omega}_q \end{bmatrix} \tag{2.179}$$

and show, compared to the standard FFoR approach, the absence of the forces due to the choice of reference frame, denoted by $\widetilde{\boldsymbol{\omega}}_i \boldsymbol{v}_i$ in [67].
The evaluation of the integrals of eq. (2.163) – (2.166) yields terms mostly identical to the terms of the standard FFoR approach. Coriolis and centrifugal forces $\boldsymbol{h}_\omega$ are simplified since the forces due to the choice of reference frame vanish while $\boldsymbol{M}_{rt}$ and $\boldsymbol{M}_{et}$ are post-multiplied by $\boldsymbol{A}_i$ due to different reference frames for translational and rotational components.
The resulting components dependent on elastic coordinates are summarized in table 2.1. In [57, 67] the elastic properties are calculated for the FFoR approach by evaluation of eq. (2.163) – eq. (2.166) for beam and finite element structures.

2.4 Constraints in FMBS

Algebraic constraints as shown in section 2.1.7 can be used to limit the movement between flexible bodies as well. In [57] the constraint equations are formulated relative to a joint-fixed coordinate system and using the FFoR approach. However, since the usage of quaternions as a rotation parametrization allows an efficient formulation of constraint equations relative to the global reference system, the constraint equations are derived for flexible bodies and rotation parametrization with quaternions in the following. The first constraint to be

Table 2.1: Summary of SID-structure.

Type	Content
Nodal information	$\boldsymbol{R}_k$: origin of node k $\boldsymbol{\Phi}_k$: translational elasticity of k $\boldsymbol{\Psi}_k$: rotational elasticity of k
Mass matrix	m_i: mass of body i $\boldsymbol{c}_i$: CoG of body i $\boldsymbol{C}_i^t$: translational-elastic coupling $\boldsymbol{C}_i^r$: rotational-elastic coupling $\boldsymbol{M}_i^e$: mass matrix of elastic coordinates $\boldsymbol{q}_i$ $\boldsymbol{I}_i$: inertia tensor
Coriolis- and centrifugal forces	$\boldsymbol{G}_i^r$: Coriolis forces, rotational components $\boldsymbol{G}_i^e$: Coriolis forces, elastic components $\boldsymbol{O}_i^e$: centrifugal forces of el. coordinates
Inner forces	$\boldsymbol{K}_i^e$: internal stiffness matrix $\boldsymbol{D}_i^e$: internal damping matrix

calculated is the ball constraint, coupling all three translational degrees of freedom of two body points. Afterwards, the rotational joint and connection rod are introduced. These constraints are sufficient to model the suspension kinematics of a road vehicle with the common McPherson type suspension but also many other suspension configurations.
The system of DAEs from eq. (2.96) consider the constraint equations on position level, forming a index-3 set of DAEs. To gain the constraint equations on velocity and acceleration level necessary for real-time capable solution of the MBS, the constraint equations on velocity and acceleration level are derived as well. The mixed reference frame approach from section 2.3.2 is used here. When using the standard FFoR this will result in slightly different equations since the position vectors ${}^i\boldsymbol{\rho}_i, {}^i\boldsymbol{\rho}_j$ have to be transformed into the global reference system.

2.4.1 Ball Joint

A ball constraint as described in eq. (2.75) is defined for flexible bodies as

$$\boldsymbol{g} = {}^I\boldsymbol{\rho}_i + \boldsymbol{A}_i^T \underbrace{\left({}^i\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)}_{\boldsymbol{r}_k} - {}^I\boldsymbol{\rho}_j - \boldsymbol{A}_j^T \underbrace{\left({}^i\boldsymbol{R}_l + \boldsymbol{\Phi}_l \boldsymbol{q}_j\right)}_{\boldsymbol{r}_l} = \boldsymbol{0} \tag{2.180}$$

with the linear deformation approach from eq. (2.142). Calculation of the time-differentiation of eq. (2.180) yields to

$$\frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}t} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \boldsymbol{0} \tag{2.181}$$

and

$$\frac{\mathrm{d}^2\boldsymbol{g}}{\mathrm{d}t^2} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\ddot{\boldsymbol{x}} = \boldsymbol{0}\,. \tag{2.182}$$

Hereby $\boldsymbol{x}$ denotes the vector of all coordinates of the multibody system with $\boldsymbol{x} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_n]$, where only the entries of the corresponding bodies i, j are necessary for the calculation of the constraint equations.
To calculate the Jacobian of $\boldsymbol{g}$, it has to be derived with respect to all coordinates of the MBS. Since for cartesian coordinates the constraint formulation is only dependent on the coordinates of bodies i, j, the Jacobian is

$$\boldsymbol{G} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} & \cdots & \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_i} & \boldsymbol{0} & \cdots & \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_j} & \boldsymbol{0} & \cdots \end{bmatrix} \tag{2.183}$$

and only two entries are non-zero. The Jacobian with respect to $\boldsymbol{x}_i, \boldsymbol{x}_j$ then yields with eq. (2.41)

$$\begin{aligned} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_i} &= \begin{bmatrix} \boldsymbol{E} & 2\boldsymbol{G}_i\bar{\boldsymbol{r}}_k + 2\boldsymbol{r}_k\boldsymbol{p}_i^T & \boldsymbol{A}_i^T\boldsymbol{\Phi}_k \end{bmatrix} \\ \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_j} &= \begin{bmatrix} -\boldsymbol{E} & -2\boldsymbol{G}_j\bar{\boldsymbol{r}}_l - 2\boldsymbol{r}_l\boldsymbol{p}_j^T & -\boldsymbol{A}_j^T\boldsymbol{\Phi}_l \end{bmatrix} \end{aligned} \tag{2.184}$$

The form $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \boldsymbol{0}$ states the constraint equation on velocity level by using $\dot{\boldsymbol{x}} = [\boldsymbol{v}\ \dot{\boldsymbol{p}}\ \boldsymbol{q}]^T$. To state the constraint equation using the velocity vector $\boldsymbol{z} = [\boldsymbol{v}\ \boldsymbol{\omega}\ \boldsymbol{q}]^T$, eq. (2.184) is transformed with eq. (2.71) as

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\boldsymbol{Z}\boldsymbol{z} = \boldsymbol{H}\boldsymbol{z} = \boldsymbol{0}\,. \tag{2.185}$$

With eq. (2.33) follows for eq. (2.184)

$$\begin{aligned} \boldsymbol{H}_i &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{G}_i\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T & \boldsymbol{A}_i^T\boldsymbol{\Phi}_k \end{bmatrix} \\ \boldsymbol{H}_j &= \begin{bmatrix} -\boldsymbol{E} & -\boldsymbol{G}_j\bar{\boldsymbol{r}}_l\boldsymbol{L}_j^T & -\boldsymbol{A}_j^T\boldsymbol{\Phi}_l \end{bmatrix} \end{aligned} \tag{2.186}$$

and by using eq. (2.37), eq. (2.23) to

$$\begin{aligned} \boldsymbol{H}_i &= \begin{bmatrix} \boldsymbol{E} & -\boldsymbol{A}_i^T\widetilde{\boldsymbol{r}}_k & \boldsymbol{A}_i^T\boldsymbol{\Phi}_k \end{bmatrix} \\ \boldsymbol{H}_j &= \begin{bmatrix} -\boldsymbol{E} & \boldsymbol{A}_j^T\widetilde{\boldsymbol{r}}_l & -\boldsymbol{A}_j^T\boldsymbol{\Phi}_l \end{bmatrix}. \end{aligned} \tag{2.187}$$

The second part of eq. (2.182) can be moved to the left side of the equation and defines the right-hand side of the constraint equation on acceleration level,

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\ddot{\boldsymbol{x}} = -\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}} = \boldsymbol{\gamma} \tag{2.188}$$

While the left-hand side is already known, the right-hand side ($\boldsymbol{\gamma}$) has to be calculated in additional steps. To complete this step, the term $-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}}$ is transformed with the chain rule to

$$-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}} = -\left(\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}}\right)\dot{\boldsymbol{x}}\,. \tag{2.189}$$

By considering that $\frac{\partial}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \boldsymbol{0}$, eq. (2.189) can also be written as

$$-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\dot{\boldsymbol{x}} = -\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right)\dot{\boldsymbol{x}} = \boldsymbol{\gamma}\,. \tag{2.190}$$

Thus, eq. (2.184) multiplied with $\dot{\boldsymbol{x}}$ yields with eq. (2.33)

$$\begin{aligned}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \;& \boldsymbol{v}_i + 2\boldsymbol{G}_i\bar{\boldsymbol{r}}_k\dot{\boldsymbol{p}}_i + 2\boldsymbol{r}_k\boldsymbol{p}_i^T\dot{\boldsymbol{p}}_i + \boldsymbol{A}_i^T\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i \\ & - \boldsymbol{v}_j - 2\boldsymbol{G}_j\bar{\boldsymbol{r}}_l\dot{\boldsymbol{p}}_j - 2\boldsymbol{r}_l\boldsymbol{p}_j^T\dot{\boldsymbol{p}}_j - \boldsymbol{A}_j^T\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j \\ = \;& \boldsymbol{v}_i + 2\boldsymbol{G}_i\bar{\boldsymbol{r}}_k\dot{\boldsymbol{p}}_i + \boldsymbol{A}_i^T\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i - \boldsymbol{v}_j - 2\boldsymbol{G}_j\bar{\boldsymbol{r}}_l\dot{\boldsymbol{p}}_j - \boldsymbol{A}_j^T\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\,.\end{aligned} \tag{2.191}$$

When calculating the derivative $\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right)$, the resultant matrix is again only dependent on the coordinates of bodies i, j with the non-zero entries

$$\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right) = \begin{bmatrix}\boldsymbol{0} & \cdots & \frac{\partial}{\partial \boldsymbol{\rho}_i} & \frac{\partial}{\partial \boldsymbol{p}_i} & \frac{\partial}{\partial \boldsymbol{q}_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \frac{\partial}{\partial \boldsymbol{\rho}_j} & \frac{\partial}{\partial \boldsymbol{p}_j} & \frac{\partial}{\partial \boldsymbol{q}_j} & \cdots & \boldsymbol{0}\end{bmatrix}\,. \tag{2.192}$$

The necessary entries are

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\rho}_i} &= \boldsymbol{0} \\ \frac{\partial}{\partial \boldsymbol{p}_i} &= 2\dot{\boldsymbol{G}}_i\bar{\boldsymbol{r}}_k + 2\boldsymbol{G}_i\overline{\left(\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i\right)} + 2\left(\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i\right)\boldsymbol{p}_i^T \\ \frac{\partial}{\partial \boldsymbol{q}_i} &= 2\dot{\boldsymbol{G}}_i\boldsymbol{L}_i^T\boldsymbol{\Phi}_k \\ \frac{\partial}{\partial \boldsymbol{\rho}_j} &= \boldsymbol{0} \\ \frac{\partial}{\partial \boldsymbol{p}_j} &= -2\dot{\boldsymbol{G}}_j\bar{\boldsymbol{r}}_l - 2\boldsymbol{G}_j\overline{\left(\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right)} - 2\left(\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right)\boldsymbol{p}_j^T \\ \frac{\partial}{\partial \boldsymbol{q}_j} &= -2\dot{\boldsymbol{G}}_j\boldsymbol{L}_j^T\boldsymbol{\Phi}_l\,.\end{aligned}$$

Post-multiplying eq. (2.192) with $\dot{\boldsymbol{x}}$ and using eq. (2.38), eq. (2.33), eq. (2.30) and eq. (2.56) results for $\boldsymbol{\gamma}$

$$\begin{aligned}\boldsymbol{\gamma} = -\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right)\dot{\boldsymbol{x}} = \;& -2\dot{\boldsymbol{G}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k - 4\boldsymbol{G}_i\dot{\boldsymbol{L}}_i^T\left(\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i\right) \\ & + 2\dot{\boldsymbol{G}}_j\dot{\boldsymbol{L}}_j^T\boldsymbol{r}_l + 4\boldsymbol{G}_j\dot{\boldsymbol{L}}_j^T\left(\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right) \\ = \;& -2\dot{\boldsymbol{G}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k + 2\dot{\boldsymbol{G}}_j\dot{\boldsymbol{L}}_j^T\boldsymbol{r}_l \\ & - 2\boldsymbol{A}_i^T\widetilde{\boldsymbol{\omega}}_i\left(\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i\right) + 2\boldsymbol{A}_j^T\widetilde{\boldsymbol{\omega}}_j\left(\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right)\,.\end{aligned} \tag{2.193}$$

Eq. (2.188) defines the constraint equation on acceleration level using $\ddot{\boldsymbol{x}} = [\boldsymbol{a}\ \ddot{\boldsymbol{p}}]^T$. To state the equation as a function of $\dot{\boldsymbol{\omega}}$, eq. (2.32) is inserted in eq. (2.193) and simplified using eq. (2.16) eq. (2.36) and eq. (2.19). This results in

$$\begin{aligned}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{\omega}} &= \boldsymbol{\gamma} + 2\boldsymbol{G}_i\bar{\boldsymbol{r}}_k\frac{1}{4}\omega_i^2\boldsymbol{p}_i - 2\boldsymbol{G}_j\bar{\boldsymbol{r}}_l\frac{1}{4}\omega_j^2\boldsymbol{p}_j + 2\boldsymbol{r}_k\boldsymbol{p}_i^T\frac{1}{4}\omega_i^2\boldsymbol{p}_i - 2\boldsymbol{r}_l\boldsymbol{p}_j^T\frac{1}{4}\omega_j^2\boldsymbol{p}_j \\ &= \boldsymbol{\gamma} + 2\boldsymbol{G}_i\boldsymbol{L}_i^T\boldsymbol{r}_k\frac{1}{4}\omega_i^2 - 2\boldsymbol{G}_j\boldsymbol{L}_i^T\boldsymbol{r}_l\frac{1}{4}\omega_j^2 + \frac{1}{2}\boldsymbol{r}_k\omega_i^2 - \frac{1}{2}\boldsymbol{r}_l\omega_j^2 \\ &= \boldsymbol{\gamma} + \frac{1}{2}\boldsymbol{A}_i^T\boldsymbol{r}_k\omega_i^2 - \frac{1}{2}\boldsymbol{A}_j^T\boldsymbol{r}_l\omega_j^2 + \frac{1}{2}\boldsymbol{r}_k\omega_i^2 - \frac{1}{2}\boldsymbol{r}_l\omega_j^2\,. \end{aligned} \tag{2.194}$$

2.4.2 Rotational Constraint

Using the rotational constraint definition of eq. (2.79), the constraint is defined by limiting the relative rotational motion in addition to the translational motion described by the equations of a ball joint. Consider the local rotation axes at the points k, l as ${}^k\boldsymbol{w}_k$ and ${}^l\boldsymbol{w}_l$, the rotational constraint is defined by setting the cross product to zero. Using eq. (2.143) this results in two vectors in global coordinates,

$${}^I\boldsymbol{w}_k = \boldsymbol{A}_i^T\boldsymbol{D}_k^T{}^k\boldsymbol{w}_k = \boldsymbol{A}_i^T\underbrace{\boldsymbol{\Gamma}_k^T\left(\boldsymbol{E} + \widetilde{\boldsymbol{\Psi}_k\boldsymbol{q}_i}\right){}^k\boldsymbol{w}_k}_{{}^i\boldsymbol{w}_k} \tag{2.195}$$

$${}^I\boldsymbol{w}_l = \boldsymbol{A}_j^T\boldsymbol{D}_l^T{}^l\boldsymbol{w}_l = \boldsymbol{A}_j^T\underbrace{\boldsymbol{\Gamma}_l^T\left(\boldsymbol{E} + \widetilde{\boldsymbol{\Psi}_l\boldsymbol{q}_j}\right){}^l\boldsymbol{w}_l}_{{}^j\boldsymbol{w}_l}\,. \tag{2.196}$$

A rotational constraint for flexible bodies then can be defined analog to eq. (2.79) as

$$\boldsymbol{g} = {}^I\widetilde{\boldsymbol{w}}_k{}^I\boldsymbol{w}_l = \boldsymbol{0}\,. \tag{2.197}$$

The Jacobian of eq. (2.197) yields with eq. (2.41) and the identity ${}^I\widetilde{\boldsymbol{w}}_k{}^I\boldsymbol{w}_l = -{}^I\widetilde{\boldsymbol{w}}_l{}^I\boldsymbol{w}_k$ to

$$\begin{aligned}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_i} &= \begin{bmatrix}\boldsymbol{0} & -{}^I\widetilde{\boldsymbol{w}}_l\left(2\boldsymbol{G}_i{}^i\overline{\boldsymbol{w}}_k + 2{}^i\boldsymbol{w}_k\boldsymbol{p}_i^T\right) & {}^I\widetilde{\boldsymbol{w}}_l\boldsymbol{A}_i^T\boldsymbol{\Gamma}_k^T{}^k\widetilde{\boldsymbol{w}}_k\boldsymbol{\Psi}_k\end{bmatrix} \\ \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}_j} &= \begin{bmatrix}\boldsymbol{0} & {}^I\widetilde{\boldsymbol{w}}_k\left(2\boldsymbol{G}_j{}^j\overline{\boldsymbol{w}}_l + 2{}^j\boldsymbol{w}_l\boldsymbol{p}_j^T\right) & -{}^I\widetilde{\boldsymbol{w}}_k\boldsymbol{A}_j^T\boldsymbol{\Gamma}_l{}^l\widetilde{\boldsymbol{w}}_l\boldsymbol{\Psi}_l\end{bmatrix}\,. \end{aligned} \tag{2.198}$$

Similar to eq. (2.187), it is necessary to express eq. (2.198) in terms of angular velocity, which results in

$$\begin{aligned}\boldsymbol{H}_i &= \begin{bmatrix}\boldsymbol{0} & {}^I\widetilde{\boldsymbol{w}}_l\boldsymbol{A}_i^T{}^i\widetilde{\boldsymbol{w}}_k & {}^I\widetilde{\boldsymbol{w}}_l\boldsymbol{A}_i^T\boldsymbol{\Gamma}_k^T{}^k\widetilde{\boldsymbol{w}}_k\boldsymbol{\Psi}_k\end{bmatrix} \\ \boldsymbol{H}_j &= \begin{bmatrix}\boldsymbol{0} & -{}^I\widetilde{\boldsymbol{w}}_k\boldsymbol{A}_j^T{}^j\widetilde{\boldsymbol{w}}_l & -{}^I\widetilde{\boldsymbol{w}}_k\boldsymbol{A}_j^T\boldsymbol{\Gamma}_l{}^l\widetilde{\boldsymbol{w}}_l\boldsymbol{\Psi}_l\end{bmatrix}\,. \end{aligned} \tag{2.199}$$

Derivation of the components $\boldsymbol{\gamma}$ is performed using the same steps as for eq. (2.193) and eq. (2.194).

First, $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}$ is calculated using eq. (2.33).

$$\begin{aligned}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} =& - {}^{I}\widetilde{\boldsymbol{w}}_l \left(2\boldsymbol{G}_i {}^{i}\overline{\boldsymbol{w}}_k + 2\boldsymbol{r}_k \boldsymbol{p}_i^T\right)\dot{\boldsymbol{p}}_i + {}^{I}\widetilde{\boldsymbol{w}}_l \boldsymbol{A}_i^T \boldsymbol{\Gamma}_k^{T\,k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \dot{\boldsymbol{q}}_i \\ &+ {}^{I}\widetilde{\boldsymbol{w}}_k \left(2\boldsymbol{G}_j {}^{j}\overline{\boldsymbol{w}}_l + 2\boldsymbol{r}_l \boldsymbol{p}_j^T\right)\dot{\boldsymbol{p}}_j - {}^{I}\widetilde{\boldsymbol{w}}_k \boldsymbol{A}_j^T \boldsymbol{\Gamma}_l^{T\,l}\widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l \dot{\boldsymbol{q}}_j \\ =& {}^{I}\widetilde{\boldsymbol{w}}_l \Big(\boldsymbol{A}_i^T \underbrace{\boldsymbol{\Gamma}_k^{T\,k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \dot{\boldsymbol{q}}_i}_{\boldsymbol{u}_k} - 2\boldsymbol{G}_i {}^{i}\overline{\boldsymbol{w}}_k \dot{\boldsymbol{p}}_i\Big) - {}^{I}\widetilde{\boldsymbol{w}}_k \Big(\boldsymbol{A}_j^T \underbrace{\boldsymbol{\Gamma}_l^{T\,l}\widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l \dot{\boldsymbol{q}}_j}_{\boldsymbol{u}_l} - 2\boldsymbol{G}_j {}^{j}\overline{\boldsymbol{w}}_l \dot{\boldsymbol{p}}_j\Big)\end{aligned} \tag{2.200}$$

and taking the derivative $\frac{\partial}{\partial \boldsymbol{x}}$ yields to the entries for matrix (2.192)

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\rho}_i} &= \boldsymbol{0} \\ \frac{\partial}{\partial \boldsymbol{p}_i} &= {}^{I}\widetilde{\boldsymbol{w}}_l \left(2\boldsymbol{G}_i \bar{\boldsymbol{u}}_k + 2\boldsymbol{u}_k \boldsymbol{p}_i^T - 2\dot{\boldsymbol{G}}_i {}^{i}\overline{\boldsymbol{w}}_k\right) \\ &\quad + \left(\boldsymbol{A}_j^T \boldsymbol{u}_l - 2\boldsymbol{G}_j {}^{j}\overline{\boldsymbol{w}}_l \dot{\boldsymbol{p}}_j\right)\widetilde{\ } \left(2\boldsymbol{G}_i {}^{i}\overline{\boldsymbol{w}}_k + 2{}^{i}\boldsymbol{w}_k \boldsymbol{p}_i^T\right) \\ \frac{\partial}{\partial \boldsymbol{q}_i} &= 2{}^{I}\widetilde{\boldsymbol{w}}_l \boldsymbol{G}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{\Gamma}_k{}^{k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \\ &\quad - \left(\boldsymbol{A}_j^T \boldsymbol{u}_l - 2\boldsymbol{G}_j {}^{j}\overline{\boldsymbol{w}}_l \dot{\boldsymbol{p}}_j\right)\widetilde{\ } \boldsymbol{A}_i \boldsymbol{\Gamma}_k^{T\,k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \\ \frac{\partial}{\partial \boldsymbol{\rho}_j} &= \boldsymbol{0} \\ \frac{\partial}{\partial \boldsymbol{p}_j} &= -{}^{I}\widetilde{\boldsymbol{w}}_k \left(2\boldsymbol{G}_j \bar{\boldsymbol{u}}_l + 2\boldsymbol{u}_l \boldsymbol{p}_j^T - 2\dot{\boldsymbol{G}}_j {}^{j}\overline{\boldsymbol{w}}_l\right) \\ &\quad - \left(\boldsymbol{A}_i^T \boldsymbol{u}_k - 2\boldsymbol{G}_i {}^{i}\overline{\boldsymbol{w}}_k \dot{\boldsymbol{p}}_i\right)\widetilde{\ } \left(2\boldsymbol{G}_j {}^{j}\overline{\boldsymbol{w}}_l + 2{}^{j}\boldsymbol{w}_l \boldsymbol{p}_j^T\right) \\ \frac{\partial}{\partial \boldsymbol{q}_j} &= -2{}^{I}\widetilde{\boldsymbol{w}}_k \boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T \boldsymbol{\Gamma}_l{}^{l}\widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l \\ &\quad + \left(\boldsymbol{A}_i^T \boldsymbol{u}_k - 2\boldsymbol{G}_i {}^{i}\overline{\boldsymbol{w}}_k \dot{\boldsymbol{p}}_i\right)\widetilde{\ } \boldsymbol{A}_j \boldsymbol{\Gamma}_l^{T\,l}\widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l\end{aligned}$$

Multiplying with $\dot{\boldsymbol{x}}$ then results in

$$\begin{aligned}\boldsymbol{\gamma} =& - \frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right)\dot{\boldsymbol{x}} \\ =& - {}^{I}\widetilde{\boldsymbol{w}}_l \left(2\boldsymbol{G}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{u}_k - 2\dot{\boldsymbol{G}}_i \dot{\boldsymbol{L}}_i^T {}^{i}\boldsymbol{w}_k\right) - \left(\boldsymbol{A}_j^T \boldsymbol{u}_l - 2\boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T {}^{j}\boldsymbol{w}_l\right)\widetilde{\ } \left(2\boldsymbol{G}_i \dot{\boldsymbol{L}}_i^T {}^{i}\boldsymbol{w}_k\right) \\ &- 2{}^{I}\widetilde{\boldsymbol{w}}_l \boldsymbol{G}_i \dot{\boldsymbol{L}}_i^T \underbrace{\boldsymbol{\Gamma}_k^{T\,k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \dot{\boldsymbol{q}}_i}_{\boldsymbol{u}_k} + \left(\boldsymbol{A}_j^T \boldsymbol{u}_l - 2\boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T {}^{j}\boldsymbol{w}_l\right)\widetilde{\ } \boldsymbol{A}_i^T \underbrace{\boldsymbol{\Gamma}_k^{T\,k}\widetilde{\boldsymbol{w}}_k \boldsymbol{\Psi}_k \dot{\boldsymbol{q}}_i}_{\boldsymbol{u}_k} \\ &+ {}^{I}\widetilde{\boldsymbol{w}}_k \left(2\boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T \boldsymbol{u}_l - 2\dot{\boldsymbol{G}}_j \dot{\boldsymbol{L}}_j^T {}^{j}\boldsymbol{w}_l\right) + \left(\boldsymbol{A}_i^T \boldsymbol{u}_k - 2\boldsymbol{G}_i \dot{\boldsymbol{L}}_i^T {}^{i}\boldsymbol{w}_k\right)\widetilde{\ } \left(2\boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T {}^{j}\boldsymbol{w}_l\right)\end{aligned}$$

$$
\begin{aligned}
&+ 2\,{}^I\widetilde{\boldsymbol{w}}_k \boldsymbol{G}_j \dot{\boldsymbol{L}}_j^T \underbrace{\boldsymbol{\Gamma}_l^{Tl} \widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l \dot{\boldsymbol{q}}_j}_{\boldsymbol{u}_l} - \left(\boldsymbol{A}_i^T \boldsymbol{u}_k - 2\boldsymbol{G}_i \dot{\boldsymbol{L}}_i^{T\,i} \boldsymbol{w}_k\right)\widetilde{} \boldsymbol{A}_j^T \underbrace{\boldsymbol{\Gamma}_l^{Tl} \widetilde{\boldsymbol{w}}_l \boldsymbol{\Psi}_l \dot{\boldsymbol{q}}_j}_{\boldsymbol{u}_l} \\
&= -{}^I\widetilde{\boldsymbol{w}}_l \left(2\boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i \boldsymbol{u}_k - 2\dot{\boldsymbol{G}}_i \dot{\boldsymbol{L}}_i^{T\,i} \boldsymbol{w}_k\right) + {}^I\widetilde{\boldsymbol{w}}_k \left(2\boldsymbol{A}_j^T \widetilde{\boldsymbol{\omega}}_j \boldsymbol{u}_l - 2\dot{\boldsymbol{G}}_j \dot{\boldsymbol{L}}_j^{T\,j} \boldsymbol{w}_l\right) \\
&\quad + 2\left(\boldsymbol{A}_j^T \boldsymbol{u}_l - \boldsymbol{A}_j^T \widetilde{\boldsymbol{\omega}}_j{}^j \boldsymbol{w}_l\right)\widetilde{} \left(\boldsymbol{A}_i^T \widetilde{\boldsymbol{\omega}}_i{}^i \boldsymbol{w}_k - \boldsymbol{A}_i^T \boldsymbol{u}_k\right). \qquad (2.201)
\end{aligned}
$$

The calculation of $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{\omega}}$ is done similar to eq. (2.194) with

$$
\begin{aligned}
\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{\omega}} &= \boldsymbol{\gamma} - {}^I\widetilde{\boldsymbol{w}}_l \left(2\boldsymbol{G}_i{}^i\overline{\boldsymbol{w}}_k + 2{}^i\boldsymbol{w}_k \boldsymbol{p}_i^T\right)\frac{1}{4}\omega_i^2 \boldsymbol{p}_i + {}^I\widetilde{\boldsymbol{w}}_k \left(2\boldsymbol{G}_j{}^j\overline{\boldsymbol{w}}_l + 2{}^j\boldsymbol{w}_l \boldsymbol{p}_j^T\right)\frac{1}{4}\omega_j^2 \boldsymbol{p}_j \\
&= \boldsymbol{\gamma} - \frac{1}{2}\omega_i^2 {}^I\widetilde{\boldsymbol{w}}_l \boldsymbol{G}_i{}^i\overline{\boldsymbol{w}}_k \boldsymbol{p}_i + \frac{1}{2}\omega_j^2 {}^I\widetilde{\boldsymbol{w}}_k \boldsymbol{G}_j{}^j\overline{\boldsymbol{w}}_l \boldsymbol{p}_j - \frac{1}{2}{}^I\widetilde{\boldsymbol{w}}_l{}^i\boldsymbol{w}_k \omega_i^2 + \frac{1}{2}{}^I\widetilde{\boldsymbol{w}}_k{}^j\boldsymbol{w}_l \omega_j^2 \\
&= \boldsymbol{\gamma} - \frac{1}{2}\omega_i^2 {}^I\widetilde{\boldsymbol{w}}_l \boldsymbol{A}_i^{T\,i}\boldsymbol{w}_k + \frac{1}{2}\omega_j^2 {}^I\widetilde{\boldsymbol{w}}_k \boldsymbol{A}_j^{T\,j}\boldsymbol{w}_l - \frac{1}{2}{}^I\widetilde{\boldsymbol{w}}_l{}^i\boldsymbol{w}_k \omega_i^2 + \frac{1}{2}{}^I\widetilde{\boldsymbol{w}}_k{}^j\boldsymbol{w}_l \omega_j^2 .
\end{aligned} \qquad (2.202)
$$

2.4.3 Connection Rod

A connection rod is used to constrain two points to a fixed distance. It is also called spherical-spherical joint or, depending on the field of use, control arm, steering rod or coupling rod. It has a broad variety of applications especially in vehicle dynamics, many suspension parts like steering rods, stabilizer bars or trailing arms can be modeled with this constraint. The connection rod is assumed to be a massless, rigid link. Figure 2.18 shows the constraint between two points with the distance vector $\boldsymbol{d}$.

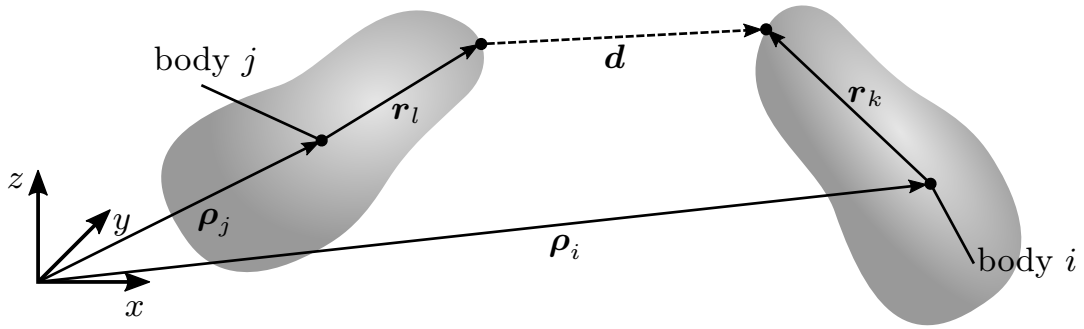


Figure 2.18: Connection rod between two bodies.

The constraint equation is stated as

$$
g = \boldsymbol{d}^T \boldsymbol{d} - l^2 = 0\,, \quad l > 0 \qquad (2.203)
$$

with length l to be constant and non-zero and $\boldsymbol{d}$ defined as

$$\begin{aligned}\boldsymbol{d} &= {}^{I}\boldsymbol{\rho}_k - {}^{I}\boldsymbol{\rho}_l \\ &= {}^{I}\boldsymbol{\rho}_i + \boldsymbol{A}_i^T \underbrace{\left({}^{i}\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)}_{\boldsymbol{r}_k} - {}^{I}\boldsymbol{\rho}_j - \boldsymbol{A}_j^T \underbrace{\left({}^{j}\boldsymbol{R}_l + \boldsymbol{\Phi}_l \boldsymbol{q}_j\right)}_{\boldsymbol{r}_l} .\end{aligned} \tag{2.204}$$

It is a scalar constraint, removing one DOF from the system. With eq. (2.41) the Jacobian for eq. (2.203) is calculated as

$$\frac{\partial g}{\partial \boldsymbol{x}_i} = \begin{bmatrix} 2\boldsymbol{d}^T & 4\boldsymbol{d}^T \left(\boldsymbol{G}_i \bar{\boldsymbol{r}}_k + \boldsymbol{r}_k \boldsymbol{p}_i^T\right) & 2\boldsymbol{d}^T \boldsymbol{A}_i^T \boldsymbol{\Phi}_k \end{bmatrix} \tag{2.205}$$

$$\frac{\partial g}{\partial \boldsymbol{x}_j} = \begin{bmatrix} -2\boldsymbol{d}^T & -4\boldsymbol{d}^T \left(\boldsymbol{G}_j \bar{\boldsymbol{r}}_l + \boldsymbol{r}_l \boldsymbol{p}_j^T\right) & -2\boldsymbol{d}^T \boldsymbol{A}_j^T \boldsymbol{\Phi}_l \end{bmatrix} \tag{2.206}$$

and expressed in terms of angular velocity with eq. (2.33) and eq. (2.37)

$$\boldsymbol{H}_i = \begin{bmatrix} 2\boldsymbol{d}^T & 2\boldsymbol{d}^T \boldsymbol{A}_i^T \widetilde{\boldsymbol{r}}_k & 2\boldsymbol{d}^T \boldsymbol{A}_i^T \boldsymbol{\Phi}_k \end{bmatrix} \tag{2.207}$$

$$\boldsymbol{H}_j = \begin{bmatrix} -2\boldsymbol{d}^T & -2\boldsymbol{d}^T \boldsymbol{A}_j^T \widetilde{\boldsymbol{r}}_l & -2\boldsymbol{d}^T \boldsymbol{A}_j^T \boldsymbol{\Phi}_l \end{bmatrix} . \tag{2.208}$$

For the analysis on acceleration level, the terms $\boldsymbol{\gamma}$ and $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \dot{\boldsymbol{\omega}}$ are calculated in the next steps.
The term $\frac{\partial g}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}$ is calculated using eq. (2.33) as

$$\frac{\partial g}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = 2\boldsymbol{d}^T \underbrace{\left(\boldsymbol{v}_i + 2\dot{\boldsymbol{G}}_i \boldsymbol{L}_i^T \boldsymbol{r}_k + \boldsymbol{A}_i^T \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i - \boldsymbol{v}_j - 2\dot{\boldsymbol{G}}_j \boldsymbol{L}_j^T \boldsymbol{r}_l - \boldsymbol{A}_j^T \boldsymbol{\Phi}_l \dot{\boldsymbol{q}}_j\right)}_{\boldsymbol{s}} . \tag{2.209}$$

The non-zero entries of matrix (2.192) are

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\rho}_i} &= 2\boldsymbol{s}^T \\
\frac{\partial}{\partial \boldsymbol{p}_i} &= 2\boldsymbol{s}^T \left(2\boldsymbol{G}_i \bar{\boldsymbol{r}}_k + 2\boldsymbol{r}_k \boldsymbol{p}_i^T\right) + 2\boldsymbol{d}^T \left(2\dot{\boldsymbol{G}}_i \bar{\boldsymbol{r}}_k + 2\boldsymbol{G}_i \overline{(\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i)} + 2\left(\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i\right) \boldsymbol{p}_i^T\right) \\
\frac{\partial}{\partial \boldsymbol{q}_i} &= 2\boldsymbol{s}^T \boldsymbol{A}_i^T \boldsymbol{\Phi}_k + 4\boldsymbol{d}^T \dot{\boldsymbol{G}}_i \boldsymbol{L}_i^T \boldsymbol{\Phi}_k^T \\
\frac{\partial}{\partial \boldsymbol{\rho}_j} &= -2\boldsymbol{s}^T \\
\frac{\partial}{\partial \boldsymbol{p}_j} &= -2\boldsymbol{s}^T \left(2\boldsymbol{G}_j \bar{\boldsymbol{r}}_l + 2\boldsymbol{r}_l \boldsymbol{p}_j^T\right) - 2\boldsymbol{d}^T \left(2\dot{\boldsymbol{G}}_j \bar{\boldsymbol{r}}_l + 2\boldsymbol{G}_j \overline{\left(\boldsymbol{\Phi}_l \dot{\boldsymbol{q}}_j\right)} + 2\left(\boldsymbol{\Phi}_l \dot{\boldsymbol{q}}_j\right) \boldsymbol{p}_j^T\right) \\
\frac{\partial}{\partial \boldsymbol{q}_j} &= -2\boldsymbol{s}^T \boldsymbol{A}_j^T \boldsymbol{\Phi}_l - 4\boldsymbol{d}^T \dot{\boldsymbol{G}}_j \boldsymbol{L}_j^T \boldsymbol{\Phi}_l^T .
\end{aligned}$$

Post-multiplying with $\dot{\boldsymbol{x}}$ and summarizing the terms results in

$$\begin{aligned}\boldsymbol{\gamma} &= -2\boldsymbol{s}^T\boldsymbol{s} - 4\boldsymbol{d}^T\left(\dot{\boldsymbol{G}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k + 2\dot{\boldsymbol{G}}_i\boldsymbol{L}_i^T\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{G}}_j\dot{\boldsymbol{L}}_j^T\boldsymbol{r}_l - 2\dot{\boldsymbol{G}}_j\boldsymbol{L}_j^T\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right)\\ &= -2\boldsymbol{s}^T\boldsymbol{s} - 4\boldsymbol{d}^T\left(\dot{\boldsymbol{G}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k + \boldsymbol{A}_i^T\widetilde{\boldsymbol{\omega}}_i\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i - \dot{\boldsymbol{G}}_j\dot{\boldsymbol{L}}_j^T\boldsymbol{r}_l - \boldsymbol{A}_j^T\widetilde{\boldsymbol{\omega}}_j\boldsymbol{\Phi}_l\dot{\boldsymbol{q}}_j\right)\ . \quad (2.210)\end{aligned}$$

For the constraint equations on acceleration level dependent on angular velocity follows for $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{\omega}}$ with eq. (2.32) inserted in eq. (2.207) and eq. (2.208)

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{\omega}} = \boldsymbol{\gamma} + \boldsymbol{d}^T\left(\omega_i^2\boldsymbol{A}_i^T\boldsymbol{r}_k + \omega_i^2\boldsymbol{r}_k - \omega_j^2\boldsymbol{A}_j^T\boldsymbol{r}_l - \omega_j^2\boldsymbol{r}_l\right)\ . \quad (2.211)$$

2.5 Model Order Reduction as a Method to Simulate Finite Element Models in Real-Time

Finite element methods are a common method to compute the structural deformation of large, complex structures. Using small elements with known structural dynamic behavior allow the computation of structures like a self-supporting car body by sufficiently small discretization. For dynamic analyses FE-methods result in a second order system of the form

$$\boldsymbol{M}_{FE}\ddot{\boldsymbol{q}}_{FE} + \boldsymbol{D}_{FE}\dot{\boldsymbol{q}}_{FE} + \boldsymbol{K}_{FE}\boldsymbol{q}_{FE} = \boldsymbol{f}_{FE} \quad (2.212)$$

with $\boldsymbol{M}_{FE}$, $\boldsymbol{D}_{FE}$, $\boldsymbol{K}_{FE}$ the mass, damping and stiffness matrix, $\boldsymbol{f}_{FE}$ the load vector and $\boldsymbol{q}_{FE}$ representing the vector of DOF of the FE system, see [3]. It is advantageous to use a proportional damping matrix since this will result in the same eigenvectors for the damped and the conservative system [32]. Despite having a large number of DOF, FE-models of vehicle bodies can be used for dynamic analysis like crash simulation [42] or noise, vibration and harshness (NVH) optimizations [62]. However, calculating a full FE model results in large simulation times since such models contain up to several million nodes for current models [42] with up to 6 DOF for each node. Model reduction methods have to be employed to reduce the number of DOF for real-time purposes.
A model reduction method projects a large system such as a FE model on a smaller subsystem that still represents the most important properties of the large system. Consider a solution of the full FE system approximated by

$$\boldsymbol{w}_{FE} = \boldsymbol{V}\boldsymbol{q}_{red} + \boldsymbol{w}_{res} \quad (2.213)$$

defining the $n_{FE} \times n_{red}$ projection matrix $\boldsymbol{V}$ with $n_{red} \ll n_{FE}$. This matrix defines a projection basis as a linear combination of the exact solution using n_{red} ansatz functions, represented in the columns of the projection matrix $\boldsymbol{V}$. The residuum of the approximation is denoted by $\boldsymbol{w}_{res}$. Inserted in eq. (2.212) yields

$$\boldsymbol{M}_{FE}\boldsymbol{V}\ddot{\boldsymbol{q}}_{red} + \boldsymbol{D}_{FE}\boldsymbol{V}\dot{\boldsymbol{q}}_{red} + \boldsymbol{K}_{FE}\boldsymbol{V}\boldsymbol{q}_{red} = \boldsymbol{f}_{FE} + \overline{\boldsymbol{w}}_{res} \quad (2.214)$$

with the residuum

$$\overline{\boldsymbol{w}}_{res} = -\boldsymbol{M}_{FE}\ddot{\boldsymbol{w}}_{res} - \boldsymbol{D}_{FE}\dot{\boldsymbol{w}}_{res} - \boldsymbol{K}_{FE}\boldsymbol{w}_{res} \ . \tag{2.215}$$

The Petrov-Galerkin-Condition states that for a $n_{FE} \times n_{red}$ projection matrix $\boldsymbol{W}$ the residuum vanishes with

$$\boldsymbol{W}^T \overline{\boldsymbol{w}}_{res} = 0 \, , \tag{2.216}$$

see [18, 63]. Using the Petrov-Galerkin-Condition on eq.(2.214) defines the reduced system

$$\boldsymbol{W}^T \boldsymbol{M}_{FE} \boldsymbol{V} \ddot{\boldsymbol{q}}_{red} + \boldsymbol{W}^T \boldsymbol{D}_{FE} \boldsymbol{V} \dot{\boldsymbol{q}}_{red} + \boldsymbol{W}^T \boldsymbol{K}_{FE} \boldsymbol{V} \boldsymbol{q}_{red} = \boldsymbol{W}^T \boldsymbol{f}_{FE} \tag{2.217}$$

with the $n_{red} \times n_{red}$ reduced mass, damping and stiffness matrices $\boldsymbol{M}_e = \boldsymbol{W}^T \boldsymbol{M}_{FE} \boldsymbol{V}$, $\boldsymbol{D}_e = \boldsymbol{W}^T \boldsymbol{D}_{FE} \boldsymbol{V}$ and $\boldsymbol{K}_e = \boldsymbol{W}^T \boldsymbol{K}_{FE} \boldsymbol{V}$. For many applications it is possible to choose $\boldsymbol{W} = \boldsymbol{V}$, in this case the projection is called a orthogonal projection or Galerkin-Projection.

2.5.1 Projection Methods for Vehicle Dynamics

A broad variety of projection methods exist for model order reduction of mechanical systems. Comprehensive overviews of various reduction methods are given in [18, 32, 63]. For vehicle simulations the component mode synthesis (CMS), also called Craig-Bampton-Method [12], has turned out to cover the deformation of the vehicle body sufficiently [6]. Figure 2.19 shows the body in white of a passenger car with the FFoR location shown at the CoG and the interface nodes of the front left suspension marked red.

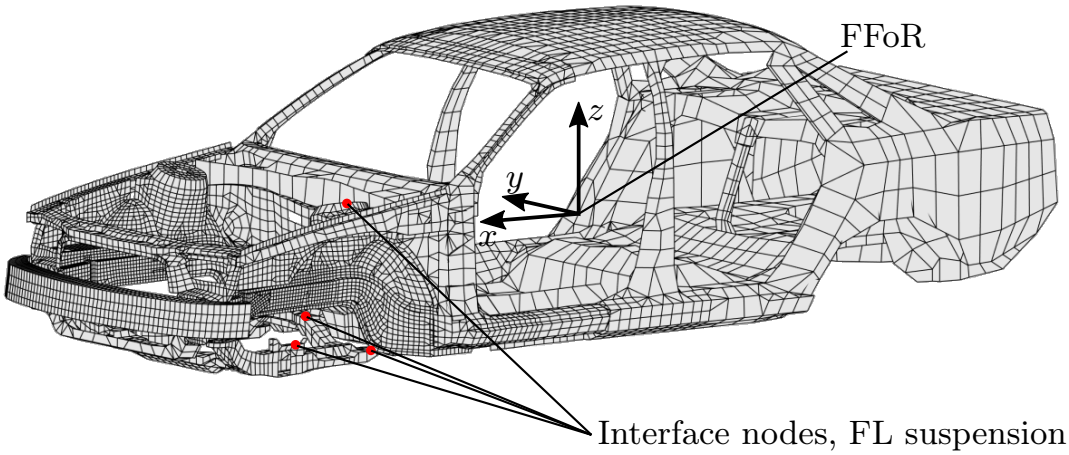


Figure 2.19: Body-in-white with interface nodes.

The interface nodes serve as a boundary condition for performing a static mode analysis of the system. Therefore, the system is partitioned into the boundary

or interface DOF and the inner DOF. To calculate the static displacement of the inner DOF, the system

$$\boldsymbol{K}_{FE}\boldsymbol{q}_{FE}^{p} = \begin{bmatrix} \boldsymbol{K}_{FE}^{ii} & \boldsymbol{K}_{FE}^{ib} \\ \boldsymbol{K}_{FE}^{bi} & \boldsymbol{K}_{FE}^{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{FE}^{i} \\ \boldsymbol{q}_{FE}^{b} \end{bmatrix} = \boldsymbol{0} \tag{2.218}$$

is solved for the upper part of eq. (2.218) with

$$\begin{aligned} \boldsymbol{0} =& \boldsymbol{K}_{FE}^{ii}\boldsymbol{q}_{FE}^{i} + \boldsymbol{K}_{FE}^{ib}\boldsymbol{q}_{FE}^{b} \\ \Rightarrow \boldsymbol{q}_{FE}^{i} =& -{\boldsymbol{K}_{FE}^{ii}}^{-1}\boldsymbol{K}_{FE}^{ib}\boldsymbol{q}_{FE}^{b} \,. \end{aligned} \tag{2.219}$$

By choosing a $n_b \times n_b$ unity matrix as the boundary condition $\boldsymbol{q}_{FE}^{b}$, all static modes can be calculated simultaneously. The resulting matrix

$$\begin{bmatrix} \boldsymbol{q}_{FE}^{i} \\ \boldsymbol{E}_{bb} \end{bmatrix} = \boldsymbol{V}_B \tag{2.220}$$

then states the projection matrix $\boldsymbol{V}$.
Additional dynamic modes may be desired to cover the dynamic behavior of the FE body. Similar to the stiffness matrix, the mass matrix can be partitioned as

$$\boldsymbol{M}_{FE} = \begin{bmatrix} \boldsymbol{M}_{FE}^{ii} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{FE}^{bb} \end{bmatrix} \tag{2.221}$$

and an eigenvalue analysis of the undamped, bounded system

$$\boldsymbol{\lambda}\boldsymbol{M}_{FE}^{ii}\boldsymbol{v} = \boldsymbol{K}_{FE}^{ii}\boldsymbol{v} \tag{2.222}$$

results in inner modes of the bounded system. Taking a selection of n_b inner modes results in the projection matrix

$$\begin{bmatrix} \boldsymbol{q}_{FE}^{i} & \boldsymbol{v}_1 \cdots \boldsymbol{v}_{n_b} \\ \boldsymbol{E}_{bb} & \boldsymbol{0} \end{bmatrix} = \boldsymbol{V}_{CMS} \,. \tag{2.223}$$

This matrix does not represent a orthogonal basis for the projection of eq. (2.217). To gain a orthogonal basis of the reduced system, an eigenmode analysis of the undamped reduced system

$$\boldsymbol{V}_{CMS}^{T}\boldsymbol{M}_{FE}\boldsymbol{V}_{CMS}\ddot{\boldsymbol{q}}_{red} + \boldsymbol{V}_{CMS}^{T}\boldsymbol{K}_{FE}\boldsymbol{V}_{CMS}\boldsymbol{q}_{red} = \boldsymbol{0} \tag{2.224}$$

has to be performed. With the resulting eigenvectors $\boldsymbol{V}_o\,[\boldsymbol{v}_{o,1} \cdots \boldsymbol{v}_{o,n_{red}}]$ an orthogonal projection matrix

$$\boldsymbol{V} = \boldsymbol{V}_{CMS}\boldsymbol{V}_o \tag{2.225}$$

of the CMS-reduced system can be calculated. By performing a modal analysis of eq. (2.222) with the unconstrained system of eq. (2.212) a purely modal reduction is performed. However, since a modal reduction does not consider the influence of interface nodes that play a major role in MBS, their application to vehicle dynamic simulations is limited.

FLEXIBLE MULTIBODY SYSTEMS USING QUATERNION DERIVATIVES

In section 2.1.8 the kinematics and kinetics of a flexible body i are derived with respect to the state vectors

$$\boldsymbol{x}_i = \begin{bmatrix} {}^I\boldsymbol{\rho}_i \\ \boldsymbol{p}_i \\ \boldsymbol{q}_i \end{bmatrix}, \; \boldsymbol{z}_i = \begin{bmatrix} {}^I\boldsymbol{\rho}_i \\ \boldsymbol{\omega}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix} \tag{3.1}$$

which are linked by the kinematic matrix $\boldsymbol{Z}_i(\boldsymbol{x}_i)$

$$\dot{\boldsymbol{x}}_i = \boldsymbol{Z}_i \boldsymbol{z}_i \;, \tag{3.2}$$

see eq. (2.93). The equations of motion (EOM) from eq. (2.94) are stated as

$$\boldsymbol{M}_i \dot{\boldsymbol{z}}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{z}_i, t) \tag{3.3}$$

and are dependent on the state vectors $\boldsymbol{x}_i, \boldsymbol{z}_i$ as well. This approach leads to additional calculations necessary since the time derivative of a quaternion $\boldsymbol{p}_i$ is not equal to the angular velocity $\boldsymbol{\omega}_i$. To circumvent these calculations, the equations of motion also can be formulated with respect to $\boldsymbol{p}_i, \dot{\boldsymbol{p}}_i$ and the state vectors

$$\boldsymbol{x}_i = \begin{bmatrix} {}^I\boldsymbol{\rho}_i \\ \boldsymbol{p}_i \\ \boldsymbol{q}_i \end{bmatrix}, \; \dot{\boldsymbol{x}}_i = \begin{bmatrix} {}^I\dot{\boldsymbol{\rho}}_i \\ \dot{\boldsymbol{p}}_i \\ \dot{\boldsymbol{q}}_i \end{bmatrix} \;. \tag{3.4}$$

This results in EOM of the form

$$\boldsymbol{M} \ddot{\boldsymbol{x}}_i = \boldsymbol{f}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, t) \tag{3.5}$$

and transferred into state-space form with $\boldsymbol{x}_1 = \boldsymbol{x}_i$, $\boldsymbol{x}_2 = \dot{\boldsymbol{x}}_i$

$$\begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix} = \boldsymbol{F}(\boldsymbol{x}_1, \boldsymbol{x}_2, t) = \begin{bmatrix} \boldsymbol{x}_2 \\ \boldsymbol{M}_i^{-1} \boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2, t) \end{bmatrix} . \tag{3.6}$$

The upper entries on the right side are only dependent on $\boldsymbol{x}_2$. This can be very useful when calculating the Jacobian of the EOM which is necessary e.g. for the solution with implicit solvers. The Jacobian then yields

$$\frac{\partial \boldsymbol{F}}{\partial [\boldsymbol{x}_1 \; \boldsymbol{x}_2]} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{E} \\ \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_1} & \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_2} \end{bmatrix} . \tag{3.7}$$

Constraint equations on index-2 and index-1 level are already available for this type of calculation from the derivation in section 2.4.
To get the EOM solely dependent on quaternions and their derivatives, Jourdain's principle from eq. (2.149) is evaluated with respect to unit quaternions and their derivatives. First, the kinematics are derived for quaternion derivatives in the next section, then the kinetics for flexible bodies with respect to quaternion derivatives are evaluated. The resulting terms are formulated to be dependent only on the state vectors and the SID entries shown in table 2.1, which prevents a reevaluation of the integrals of Jourdain's principle. Extensive use of quaternion identities shown in section 2.1.3 is made to transform the integrals, the identities are referenced at the respective equations.

3.1 Kinematics

The position of a point k on a body i is equal to eq. (2.142) with

$$ {}^{i}\boldsymbol{\rho}_k = \boldsymbol{A}_i {}^{I}\boldsymbol{\rho}_i + \underbrace{\left({}^{i}\boldsymbol{R}_k + \boldsymbol{\Phi}_k \boldsymbol{q}_i\right)}_{\boldsymbol{r}_k} \tag{3.8} $$

Then, the velocity of a point k is derived from eq. (2.144) by successive application of eq. (2.26), eq. (2.57) and eq. (2.38) as

$$ \begin{aligned} {}^{i}\boldsymbol{v}_k &= \boldsymbol{A}_i {}^{I}\dot{\boldsymbol{\rho}}_i + \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \\ &= \boldsymbol{A}_i \dot{\boldsymbol{\rho}}_i + \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\boldsymbol{L}_i \bar{\boldsymbol{r}}_k \dot{\boldsymbol{p}}_i \\ &= \begin{bmatrix} \boldsymbol{A}_i & 2\boldsymbol{L}_i \bar{\boldsymbol{r}}_k & \boldsymbol{\Phi}_k \end{bmatrix} \dot{\boldsymbol{x}}_i \\ &= \boldsymbol{T}_k^t \, \dot{\boldsymbol{x}}_i \end{aligned} \tag{3.9} $$

and yields a similar matrix $\boldsymbol{T}_k^t$ like eq. (2.144). The acceleration of a point k then yields with eq. (2.39)

$$ \begin{aligned} {}^{i}\boldsymbol{a}_k =& \boldsymbol{A}_i {}^{I}\ddot{\boldsymbol{\rho}}_i + \boldsymbol{\Phi}_k \ddot{\boldsymbol{q}}_i + 2\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \\ &+ 2\boldsymbol{L}_i \ddot{\boldsymbol{L}}_i^T \boldsymbol{r}_k + 2\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \left(\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k\right) \\ =& \boldsymbol{T}_k^t \, \ddot{\boldsymbol{x}}_i + 4\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \\ &+ 4\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \\ =& \boldsymbol{T}_k^t \, \ddot{\boldsymbol{x}}_i + \boldsymbol{\zeta}_k^t \end{aligned} \tag{3.10} $$

with

$$ \boldsymbol{\zeta}_k^t = 4\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i + 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k + 4\boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{L}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \, . \tag{3.11} $$

For the last term of eq. (3.11) the following transformation can be used with the help of eq. (2.27) and eq. (2.28):

$$\begin{aligned}\boldsymbol{L}_i\dot{\boldsymbol{L}}_i^T\,\boldsymbol{L}_i\dot{\boldsymbol{L}}_i^T &= -\dot{\boldsymbol{L}}_i\boldsymbol{L}_i^T\,\boldsymbol{L}_i\dot{\boldsymbol{L}}_i^T \\ &= -\dot{\boldsymbol{L}}_i\left(\boldsymbol{E}-\boldsymbol{p}_i\boldsymbol{p}_i^T\right)\dot{\boldsymbol{L}}_i^T = -\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T + \dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\end{aligned} \tag{3.12}$$

By additional application of eq. (2.38) and eq. (2.37) to the first term of eq. (3.11), this simplifies eq. (3.11) to

$$\boldsymbol{\zeta}_k^t = -4\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\,. \tag{3.13}$$

3.2 Kinetics

The EOM are evaluated with Jourdain's principle equally to the process when using angular velocity and acceleration. The main difference occurs when evaluating the integrals of inertial forces. To get the mass matrix and the vector of coriolis and centrifugal forces, eq. (2.163) has to be evaluated with eq. (3.10) and eq. (3.13).
For the mass matrix follows with $\boldsymbol{T}_k^t$ from eq. (3.9)

$$\boldsymbol{M}_i = \int_{\Omega_0}\boldsymbol{T}_k^{t\,T}\boldsymbol{T}_k^t\,\mathrm{d}m = \int_{\Omega_0}\begin{bmatrix}\boldsymbol{A}_i^T\\ -2\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\\ \boldsymbol{\Phi}_k^T\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_i & 2\boldsymbol{L}_i\bar{\boldsymbol{r}}_k & \boldsymbol{\Phi}_k\end{bmatrix}\mathrm{d}m \tag{3.14}$$

and for the Coriolis and centrifugal forces

$$\begin{aligned}\boldsymbol{h}_{\omega,i} &= -\int_{\Omega_0}\boldsymbol{T}_k^{t\,T}\boldsymbol{\zeta}_k^t\,\mathrm{d}m \\ &= -\int_{\Omega_0}\begin{bmatrix}\boldsymbol{A}_i^T\\ -2\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\\ \boldsymbol{\Phi}_k^T\end{bmatrix}\left(-4\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\right)\mathrm{d}m\,.\end{aligned} \tag{3.15}$$

Two additional identities for arbitrary vectors $\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^3$ will be helpful for the evaluation of the integral terms. They can be verified by elementwise calculation.

$$\boldsymbol{a}^T\boldsymbol{b} = -\frac{1}{2}\mathrm{tr}\left(\widetilde{\boldsymbol{a}}\widetilde{\boldsymbol{b}}\right) \tag{3.16}$$

$$\boldsymbol{a}\boldsymbol{a}^T = \frac{1}{2}\mathrm{tr}\left(\widetilde{\boldsymbol{a}}\widetilde{\boldsymbol{a}}\right)\boldsymbol{E} - \widetilde{\boldsymbol{a}}\widetilde{\boldsymbol{a}}^T \tag{3.17}$$

$$\bar{\boldsymbol{a}}\bar{\boldsymbol{a}} = -\frac{1}{2}\mathrm{tr}\left(\widetilde{\boldsymbol{a}}\widetilde{\boldsymbol{a}}^T\right)\boldsymbol{E} \tag{3.18}$$

3.2.1 Mass Matrix Evaluation

Equation (3.14) gives a mass matrix with nine submatrices to evaluate. However, since the matrix is symmetric, only six components have to be calculated. The entries of the mass matrix will be evaluated seperately in the following.
Expanding the vectors of the integral of eq. (3.14) gives

$$\begin{aligned}\boldsymbol{M}_i &= \int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T\boldsymbol{A}_i & & \text{sym.} \\ -2\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{A}_i & -4\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{L}_i\bar{\boldsymbol{r}}_k & \\ \boldsymbol{\Phi}_k^T\boldsymbol{A}_i & 2\boldsymbol{\Phi}_k^T\boldsymbol{L}_i\bar{\boldsymbol{r}}_k & \boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k \end{bmatrix} \mathrm{d}m \\ &= \begin{bmatrix} \boldsymbol{M}_{tt} & & \text{sym.} \\ \boldsymbol{M}_{rt} & \boldsymbol{M}_{rr} & \\ \boldsymbol{M}_{et} & \boldsymbol{M}_{er} & \boldsymbol{M}_{ee} \end{bmatrix} \end{aligned} \tag{3.19}$$

for the six independent entries.

$\boldsymbol{M}_{tt}$ The translational part of the mass matrix is

$$\boldsymbol{M}_{tt} = \int_{\Omega_0} \boldsymbol{A}_i^T\boldsymbol{A}_i \mathrm{d}m = \int_{\Omega_0} \boldsymbol{E}\mathrm{d}m = m_i\boldsymbol{E} \tag{3.20}$$

and is of course unchanged to descriptions in other reference systems.

$\boldsymbol{M}_{rt}$ For the coupling of rotational and translational movement follows simplified with the transpose of eq. (2.37)

$$\begin{aligned}\boldsymbol{M}_{rt} &= \int_{\Omega_0} -2\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{A}_i \mathrm{d}m = \int_{\Omega_0} 2\left(\boldsymbol{L}_i^T\widetilde{\boldsymbol{r}}_k + \boldsymbol{p}_i\boldsymbol{r}_k^T\right)\boldsymbol{A}_i \mathrm{d}m \\ &= 2\boldsymbol{L}_i^T\int_{\Omega_0} \widetilde{\boldsymbol{r}}_k \mathrm{d}m\boldsymbol{A}_i + 2\boldsymbol{p}_i\int_{\Omega_0} \boldsymbol{r}_k^T \mathrm{d}m\boldsymbol{A}_i \\ &= 2m_i\boldsymbol{L}_i^T\widetilde{\boldsymbol{c}}_i\boldsymbol{A}_i + 2m_i\boldsymbol{p}_i\boldsymbol{c}_i^T\boldsymbol{A}_i\,. \end{aligned} \tag{3.21}$$

$\boldsymbol{M}_{rr}$ The center entry of the mass matrix represents the inertia tensor for flexible multibody systems in quaternions. With eq. (2.37) the integral resolves to

$$\begin{aligned}\boldsymbol{M}_{rr} &= \int_{\Omega_0} -4\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{L}_i\bar{\boldsymbol{r}}_k \mathrm{d}m \\ &= \int_{\Omega_0} 4\left(\boldsymbol{L}_i^T\widetilde{\boldsymbol{r}}_k + \boldsymbol{p}_i\boldsymbol{r}_k^T\right)\left(-\widetilde{\boldsymbol{r}}_k\boldsymbol{L}_i + \boldsymbol{r}_k\boldsymbol{p}_i^T\right)\mathrm{d}m\,. \end{aligned} \tag{3.22}$$

By expanding the braces and employing $\tilde{\boldsymbol{a}}\boldsymbol{a} = \boldsymbol{0}$ follows

$$\begin{aligned}\boldsymbol{M}_{rr} &= \int_{\Omega_0} -4\boldsymbol{L}_i^T \widetilde{\boldsymbol{r}}_k \tilde{\boldsymbol{r}}_k \boldsymbol{L}_i \mathrm{d}m + 4 \int_{\Omega_0} \boldsymbol{p}_i \boldsymbol{r}_k^T \boldsymbol{r}_k \boldsymbol{p}_i^T \mathrm{d}m \\ &= 4\boldsymbol{L}_i^T \int_{\Omega_0} \widetilde{\boldsymbol{r}}_k \widetilde{\boldsymbol{r}}_k^T \mathrm{d}m \, \boldsymbol{L}_i + 4\boldsymbol{p}_i \int_{\Omega_0} \boldsymbol{r}_k^T \boldsymbol{r}_k \mathrm{d}m \, \boldsymbol{p}_i^T \end{aligned} \tag{3.23}$$

The second integral can be transformed with eq. (3.16) to

$$\boldsymbol{M}_{rr} = 4\boldsymbol{L}_i^T \boldsymbol{I}_i \boldsymbol{L}_i + 2\mathrm{tr}\left(\boldsymbol{I}_i\right) \boldsymbol{p}_i \boldsymbol{p}_i^T \ . \tag{3.24}$$

The resulting inertia tensor similar to the approach for the 4×4 inertia tensor $\boldsymbol{I}_4$ in rigid body systems that is derived in [8]. Here, the inertia tensor in terms of quaternions for a rigid body is given as $\boldsymbol{I}_4 = 4\boldsymbol{L}^T \boldsymbol{I} \boldsymbol{L} + 2\mathrm{tr}\left(\boldsymbol{I}\right) \boldsymbol{p}\boldsymbol{p}^T$.

$\boldsymbol{M}_{et}$ The fourth entry couples the elastic motion to the translational motion of the reference system.

$$\begin{aligned}\boldsymbol{M}_{et} &= \int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{A}_i \mathrm{d}m = \int_{\Omega_0} \boldsymbol{\Phi}_k^T \mathrm{d}m \boldsymbol{A}_i \\ &= \boldsymbol{C}_i^t \boldsymbol{A}_i \end{aligned} \tag{3.25}$$

$\boldsymbol{M}_{er}$ Coupling of the elastic motion to the rotational motion is given by the fifth term. It follows with eq. (2.37)

$$\begin{aligned}\boldsymbol{M}_{er} &= \int_{\Omega_0} \boldsymbol{\Phi}_k^T 2\boldsymbol{L}_i \bar{\boldsymbol{r}}_k \mathrm{d}m = \int_{\Omega_0} 2\boldsymbol{\Phi}_k^T \left(-\widetilde{\boldsymbol{r}}_k \boldsymbol{L}_i + \boldsymbol{r}_k \boldsymbol{p}_i^T\right) \mathrm{d}m \\ &= 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \widetilde{\boldsymbol{r}}_k^T \mathrm{d}m \boldsymbol{L}_i + 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{r}_k \mathrm{d}m \boldsymbol{p}_i^T \\ &= 2\boldsymbol{C}_i^r \boldsymbol{L}_i + 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{R}_k \mathrm{d}m \boldsymbol{p}_i^T + 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k \mathrm{d}m \boldsymbol{q}_i \boldsymbol{p}_i^T \\ &= 2\boldsymbol{C}_i^r \boldsymbol{L}_i - \left(\sum_{j=1}^{3} \boldsymbol{O}_{i0,ej}^e\right) \boldsymbol{p}_i^T + 2\boldsymbol{M}_i^e \boldsymbol{q}_i \boldsymbol{p}_i^T \ , \end{aligned} \tag{3.26}$$

where eq. (3.16) is used to evaluate the second integral. The matrix $\boldsymbol{O}_{i0}^e$ denotes the constant term of matrix $\boldsymbol{O}_i^e$ used to calculate the coriolis and centrifugal forces, see table 2.1 and [67].

$\boldsymbol{M}_{ee}$ The sixth term represents the mass matrix and is independent of the rigid body motion of the elastic body. It is evaluated as

$$\begin{aligned}\boldsymbol{M}_{ee} &= \int_{\Omega_0} \boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k \mathrm{d}m \\ &= \boldsymbol{M}_i^e. \end{aligned} \tag{3.27}$$

With eq. (3.20) – eq.(3.27) the mass matrix is combined as

$$\boldsymbol{M}_i = \begin{bmatrix} m_i \boldsymbol{E} & & \text{sym.} \\ 2m_i \boldsymbol{L}_i^T \tilde{\boldsymbol{c}}_i \boldsymbol{A}_i & 4\boldsymbol{L}_i^T \boldsymbol{I}_i \boldsymbol{L}_i & \\ + 2m_i \boldsymbol{p}_i \boldsymbol{c}_i^T \boldsymbol{A}_i & + 2\mathrm{tr}\left(\boldsymbol{I}_i\right) \boldsymbol{p}_i \boldsymbol{p}_i^T & \\ & 2\boldsymbol{C}_i^r \boldsymbol{L}_i & \\ \boldsymbol{C}_i^t \boldsymbol{A}_i & + \left(\sum_{j=1}^{3} \boldsymbol{O}_{i0,ej}^e \right) \boldsymbol{p}_i^T & \boldsymbol{M}_i^e \\ & + 2\boldsymbol{M}_i^e \boldsymbol{q}_i \boldsymbol{p}_i^T & \end{bmatrix} \tag{3.28}$$

3.2.2 Coriolis and Centrifugal Forces

The vector Coriolis and centrifugal forces from eq. (3.15) results in three integrals to evaluate for the translational, rotational and elastic components.

$$\begin{aligned}\boldsymbol{h}_{\omega,i} &= -\int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T \\ -2\bar{\boldsymbol{r}}_k \boldsymbol{L}_i^T \\ \boldsymbol{\Phi}_k^T \end{bmatrix} \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \right) \mathrm{d}m \\ &= -\int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \right) \\ -2\bar{\boldsymbol{r}}_k \boldsymbol{L}_i^T \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \right) \\ \boldsymbol{\Phi}_k^T \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \right) \end{bmatrix} \mathrm{d}m \end{aligned} \tag{3.29}$$

which will be evaluated piecewise in the following steps.

$\boldsymbol{h}_{\omega t}$ For the translational components of eq. (3.29) follows

$$\begin{aligned}
\boldsymbol{h}_{\omega t} &= -\int_{\Omega_0} \boldsymbol{A}_i^T \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\right)\mathrm{d}m \\
&= 4\boldsymbol{A}_i^T\int_{\Omega_0}\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\mathrm{d}m\boldsymbol{L}_i\dot{\boldsymbol{p}}_i - 4\boldsymbol{A}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\int_{\Omega_0}\boldsymbol{r}_k\mathrm{d}m + 2\boldsymbol{A}_i^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\int_{\Omega_0}\boldsymbol{r}_k\mathrm{d}m \\
&= 4\boldsymbol{A}_i^T\widetilde{\boldsymbol{C}_i^{tT}\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i - 4m_i\boldsymbol{A}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{c}_i + 2m_i\boldsymbol{A}_i^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{c}_i\,. \qquad (3.30)
\end{aligned}$$

$\boldsymbol{h}_{\omega r}$ The rotational components are

$$\begin{aligned}
\boldsymbol{h}_{\omega r} &= -\int_{\Omega_0} -2\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\left(-4\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\right)\mathrm{d}m \\
&= -8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m + 8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m \\
&\quad - 4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m\,. \qquad (3.31)
\end{aligned}$$

and the resulting integrals are evaluated separately to express them in components of table 2.1.
For the first integral follows with eq. (2.37)

$$\begin{aligned}
&-8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m \\
=&-8\int_{\Omega_0}\left(-\boldsymbol{L}_i^T\widetilde{\boldsymbol{r}}_k - \boldsymbol{p}_i\boldsymbol{r}_k\right)\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m \\
=&8\boldsymbol{L}_i^T\int_{\Omega_0}\boldsymbol{r}_k\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\mathrm{d}m\boldsymbol{L}_i\dot{\boldsymbol{p}}_i - 8\boldsymbol{p}_i\int_{\Omega_0}\boldsymbol{r}_k\widetilde{\boldsymbol{\Phi}_k\dot{\boldsymbol{q}}_i}\mathrm{d}m\boldsymbol{L}_i\dot{\boldsymbol{p}}_i \\
=&8\boldsymbol{L}_i^T\sum_{l=1}^{n_q}\boldsymbol{G}_l^r\dot{q}_l\,\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 8\boldsymbol{p}_i\dot{\boldsymbol{q}}_i^T\int_{\Omega_0}\boldsymbol{\Phi}_k^T\widetilde{\boldsymbol{r}}_k\mathrm{d}m\boldsymbol{L}_i\dot{\boldsymbol{p}}_i \\
=&8\boldsymbol{L}_i^T\sum_{l=1}^{n_q}\boldsymbol{G}_l^r\dot{q}_l\,\boldsymbol{L}_i\dot{\boldsymbol{p}}_i + 8\boldsymbol{p}_i\dot{\boldsymbol{q}}_i^T\boldsymbol{C}_i^r\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\,. \qquad (3.32)
\end{aligned}$$

The second integral is evaluated with eq. (2.38), eq. (2.36), eq. (2.28), eq (2.24) and eq. (3.17) to

$$8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m$$

$$
\begin{aligned}
=&8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{p}_i^T\bar{\boldsymbol{r}}_k\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&-8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\boldsymbol{p}_i\boldsymbol{r}_k^T\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\boldsymbol{r}_k^T\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\left(\boldsymbol{E}-\boldsymbol{p}_i\boldsymbol{p}_i^T\right)\dot{\boldsymbol{p}}_i\boldsymbol{r}_k^T\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&-8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{p}_i\underbrace{\boldsymbol{p}_i^T\dot{\boldsymbol{p}}_i}_{=0}\boldsymbol{r}_k\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m+8\int_{\Omega_0}\bar{\boldsymbol{r}}_k\dot{\boldsymbol{p}}_i\boldsymbol{r}_k^T\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&8\dot{\boldsymbol{L}}_i^T\int_{\Omega_0}\boldsymbol{r}_k\boldsymbol{r}_k^T\mathrm{d}m\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\\
=&-8\dot{\boldsymbol{L}}_i^T\left(\boldsymbol{I}_i-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{I}_i\right)\boldsymbol{E}\right)\boldsymbol{L}_i\dot{\boldsymbol{p}}_i\ .
\end{aligned}
\tag{3.33}
$$

The third integral can be resolved with eq.(2.38), eq. (2.37), eq. (2.36) and eq. (3.18) to

$$
\begin{aligned}
&-4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m\\
=&-4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\dot{\boldsymbol{L}}_i\bar{\boldsymbol{r}}_k\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&-4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\left(\underbrace{-\widetilde{\boldsymbol{r}}_k\dot{\boldsymbol{L}}_i\dot{\boldsymbol{p}}_i}_{=\boldsymbol{0}}+\boldsymbol{r}_k\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\right)\mathrm{d}m\\
=&-4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\boldsymbol{L}_i^T\boldsymbol{r}_k\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&-4\int_{\Omega_0}\bar{\boldsymbol{r}}_k\bar{\boldsymbol{r}}_k\boldsymbol{p}_i\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\mathrm{d}m\\
=&\,2\,\mathrm{tr}\left(\int_{\Omega_0}\widetilde{\boldsymbol{r}}_k\widetilde{\boldsymbol{r}}_k^T\mathrm{d}m\right)\boldsymbol{p}_i\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\\
=&\,2\,\mathrm{tr}\left(\boldsymbol{I}_i\right)\boldsymbol{p}_i\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\ .
\end{aligned}
\tag{3.34}
$$

Merging equation (3.32), eq. (3.33) and eq. (3.34) results in

$$\begin{aligned}\boldsymbol{h}_{\omega r} =& 8\boldsymbol{L}_i^T \sum_{l=1}^{n_q} \boldsymbol{G}_l^r \dot{q}_l \, \boldsymbol{L}_i \dot{\boldsymbol{p}}_i - 8\boldsymbol{p}_i \dot{\boldsymbol{q}}_i^T \boldsymbol{C}_i^r \boldsymbol{L}_i \dot{\boldsymbol{p}}_i \\ & - 8\dot{\boldsymbol{L}}_i^T \left(\boldsymbol{I}_i - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{I}_i\right) \boldsymbol{E} \right) \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 2\,\mathrm{tr}\left(\boldsymbol{I}_i\right) \boldsymbol{p}_i \dot{\boldsymbol{p}}_i^T \dot{\boldsymbol{p}}_i \end{aligned} \tag{3.35}$$

$\boldsymbol{h}_{\omega e}$ The third term of eq. (3.29) represents the elastic components and yields

$$\begin{aligned}\boldsymbol{h}_{\omega e} &= -\int_{\Omega_0} \boldsymbol{\Phi}_k^T \left(-4\widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 4\dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k - 2\dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \right) \mathrm{d}m \\ &= 4\int_{\Omega_0} \boldsymbol{\Phi}_k^T \widetilde{\boldsymbol{\Phi}_k \dot{\boldsymbol{q}}_i} \mathrm{d}m \boldsymbol{L}_i \dot{\boldsymbol{p}}_i - 4\int_{\Omega_0} \boldsymbol{\Phi}_k^T \dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \mathrm{d}m + 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \mathrm{d}m \\ &= -2\sum_{l=1}^{n_q} \boldsymbol{G}_l^e \dot{q}_l \, \dot{\boldsymbol{L}}_i \boldsymbol{p}_i - 4\int_{\Omega_0} \boldsymbol{\Phi}_k^T \dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \mathrm{d}m + 2\int_{\Omega_0} \boldsymbol{\Phi}_k^T \dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \mathrm{d}m. \end{aligned} \tag{3.36}$$

The second integral can be transformed similar to eq. (3.33), but since the dimensions of $\boldsymbol{\Phi}_k^T$ are $n_q \times 3$ it has to be done separately for each row $\boldsymbol{\Phi}_{k,l}^T$, i.e. for each ansatz function of the elastic body. It yields

$$\begin{aligned} & - 4\int_{\Omega_0} \boldsymbol{\Phi}_{k,l}^T \dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{r}_k \mathrm{d}m \\ = & - 4\dot{\boldsymbol{p}}_i^T \boldsymbol{L}_i^T \int_{\Omega_0} \boldsymbol{\Phi}_{k,l} \boldsymbol{r}_k^T \mathrm{d}m \boldsymbol{L}_i \dot{\boldsymbol{p}}_i \\ = & -4\dot{\boldsymbol{p}}_i^T \boldsymbol{L}_i^T \left(\boldsymbol{O}_{i,l}^{ek\,T} - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{O}_{i,l}^{ek}\right) \boldsymbol{E} \right) \boldsymbol{L}_i \dot{\boldsymbol{p}}_i \end{aligned} \tag{3.37}$$

This term is not directly accessible from table 2.1, hence a transformation is made similar to [67] so that this term will result in a matrix-by-vector multiplication using the $n_q \times 6$ matrix $\boldsymbol{O}_i^e$.
With $\boldsymbol{L}_p = \boldsymbol{L}_i \dot{\boldsymbol{p}}_i$ the vector

$$\begin{aligned}\boldsymbol{\ell}_p^i = \big[& 4\boldsymbol{L}_p(1)^2 \quad 4\boldsymbol{L}_p(2)^2 \quad 4\boldsymbol{L}_p(3)^2 \\ & 4\boldsymbol{L}_p(1)\boldsymbol{L}_p(2) \quad 4\boldsymbol{L}_p(2)\boldsymbol{L}_p(3) \quad 4\boldsymbol{L}_p(1)\boldsymbol{L}_p(3)\big] \end{aligned} \tag{3.38}$$

and the matrix

$$\begin{aligned}\boldsymbol{O}_i^{eq} =&[\frac{1}{2}\left(\boldsymbol{O}_i^e(1)-\boldsymbol{O}_i^e(2)-\boldsymbol{O}_i^e(3)\right)\dots\\ &\frac{1}{2}\left(\boldsymbol{O}_i^e(2)-\boldsymbol{O}_i^e(1)-\boldsymbol{O}_i^e(3)\right)\dots\\ &\frac{1}{2}\left(\boldsymbol{O}_i^e(3)-\boldsymbol{O}_i^e(2)-\boldsymbol{O}_i^e(1)\right)\dots\\ &\boldsymbol{O}_i^e(4)\dots\\ &\boldsymbol{O}_i^e(5)\dots\\ &\boldsymbol{O}_i^e(6)]\end{aligned} \tag{3.39}$$

are defined and simplify the second integral of eq.(3.36) to $-\boldsymbol{O}_i^{eq}\boldsymbol{\ell}_p^i$.
The third integral of eq.(3.36) is evaluated similar to eq. (3.34) and yields

$$\begin{aligned}&2\int_{\Omega_0}\boldsymbol{\Phi}_k^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m\\ =&2\int_{\Omega_0}\boldsymbol{\Phi}_k^T\boldsymbol{r}_k\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\mathrm{d}m\\ =&2\int_{\Omega_0}\boldsymbol{\Phi}_k^T\boldsymbol{r}_k\mathrm{d}m\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i\ .\end{aligned} \tag{3.40}$$

This integral can be linked to the result of eq. (3.37) with the vector

$$\boldsymbol{m}_p^i=\begin{bmatrix}2\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i & 2\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i & 2\dot{\boldsymbol{p}}_i^T\dot{\boldsymbol{p}}_i & 0 & 0 & 0\end{bmatrix}^T \tag{3.41}$$

to

$$2\int_{\Omega_0}\boldsymbol{\Phi}_{k,l}^T\dot{\boldsymbol{L}}_i\dot{\boldsymbol{L}}_i^T\boldsymbol{r}_k\mathrm{d}m=\boldsymbol{O}_i^{eq}\boldsymbol{m}_p^i\ . \tag{3.42}$$

The complete term of the elastic components merged then results in

$$\boldsymbol{h}_{\omega e}=2\sum_{l=1}^{n_q}\boldsymbol{G}_l^e\dot{q}_l\,\boldsymbol{L}_i\dot{\boldsymbol{p}}_i-\boldsymbol{O}_i^{eq}\left(\boldsymbol{\ell}_p^i+\boldsymbol{m}_p^i\right)\ . \tag{3.43}$$

The quantities $\boldsymbol{h}_{\omega t}$, $\boldsymbol{h}_{\omega r}$ and $\boldsymbol{h}_{\omega e}$ give the complete set of Coriolis and centrifugal forces as

$$\boldsymbol{h}_\omega = \begin{bmatrix} 4\boldsymbol{A}_i^T \widetilde{\boldsymbol{C}_i^{t\,T} \dot{\boldsymbol{q}}_i} \boldsymbol{L}_i \dot{\boldsymbol{p}}_i - 4m_i \boldsymbol{A}_i^T \dot{\boldsymbol{L}}_i \boldsymbol{p}_i \boldsymbol{p}_i^T \dot{\boldsymbol{L}}_i^T \boldsymbol{c}_i + 2m_i \boldsymbol{A}_i^T \dot{\boldsymbol{L}}_i \dot{\boldsymbol{L}}_i^T \boldsymbol{c}_i \\ 8\boldsymbol{L}_i^T \sum_{l=1}^{n_q} \boldsymbol{G}_l^r \dot{q}_l \boldsymbol{L}_i \dot{\boldsymbol{p}}_i - 8\boldsymbol{p}_i \dot{\boldsymbol{q}}_i^T \boldsymbol{C}_i^r \boldsymbol{L}_i \dot{\boldsymbol{p}}_i \\ - 8\dot{\boldsymbol{L}}_i^T \left(\boldsymbol{I}_i - \frac{1}{2} \mathrm{tr} \boldsymbol{E} \left(\boldsymbol{I}_i \right) \right) \boldsymbol{L}_i \dot{\boldsymbol{p}}_i + 2 \,\mathrm{tr} \left(\boldsymbol{I}_i \right) \boldsymbol{p}_i \dot{\boldsymbol{p}}_i^T \dot{\boldsymbol{p}}_i \\ 2 \sum_{l=1}^{n_q} \boldsymbol{G}_l^e \dot{q}_l \boldsymbol{L}_i \dot{\boldsymbol{p}}_i - \boldsymbol{O}_i^{eq} \left(\boldsymbol{\ell}_p^i + \boldsymbol{m}_p^i \right) \end{bmatrix} \tag{3.44}$$

3.2.3 Internal and Body Forces, Surface Traction

Of the remaining integrals of eq. (2.161), the main difference when using quaternion derivatives is the matrix $\boldsymbol{T}_k^{t\,T}$ projecting the body and surface loads on points k onto the coordinates of the MBS. Internal forces have no effect on the rotational DOF and hence their evaluation is equal to the FFoR-approach and mixed reference approach.
If the body forces on the points of the bodies are constant, for example for gravitational forces, the body forces for the formulation in quaternions are with eq. (2.37)

$$\begin{aligned} \boldsymbol{h}_g = \int_{\Omega_0} \boldsymbol{T}^{t\,T} {}^i\boldsymbol{b}_0 \mathrm{d}m &= \int_{\Omega_0} \begin{bmatrix} \boldsymbol{A}_i^T \\ -2\bar{\boldsymbol{r}}_k \boldsymbol{L}_i^T \\ \boldsymbol{\Phi}_k^T \end{bmatrix} \mathrm{d}m \, {}^i\boldsymbol{g} \\ &= \begin{bmatrix} m_i \boldsymbol{A}_i^T \\ 2m_i \boldsymbol{L}_i^T \widetilde{\boldsymbol{c}}_i + 2m_i \boldsymbol{p}_i \boldsymbol{c}_i^T \\ \boldsymbol{C}_i^t \end{bmatrix} {}^i\boldsymbol{g} \,. \end{aligned} \tag{3.45}$$

The surface traction is calculated by the integral over the surface of the corresponding bodies. Since in general the surface loads are dependent on the application points on the surface, a general evaluation of eq. (2.166) is not possible and will not be evaluated in this work.

3.2.4 Application Example

The equations derived in the previous sections are used in a spatial double pendulum example to demonstrate the functionality of the quaternion derivative formulation. Two identical slender beam-shaped bodies of the dimensions

2 mm×5 mm×0.9 m are meshed with 1200 brick elements and model order reduced to 4 eigenmodes with a tangent reference frame attached to the first interface node. The upper beam is attached to the environment by a hinge joint, allowing rotation around the y-axis. The second beam is attached to the first beam by a ball joint. The time integration is started with a horizontal first beam and the second beam hanging vertically, see figure 3.1. The system is stated on index-1 level and the *MATLAB* ode23t solver for stiff systems is used to perform the time integration. The system lies in the $x-z$ plane, it is loaded by gravity and no other external forces are applied.

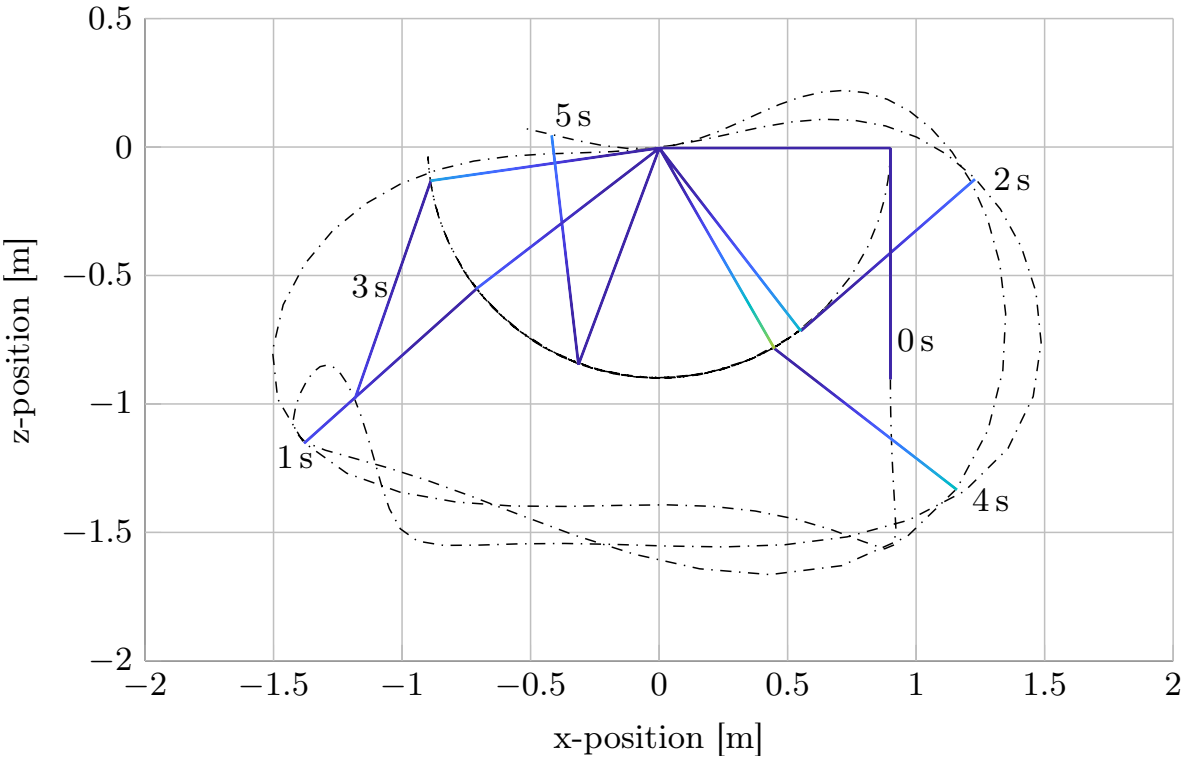


Figure 3.1: Double pendulum at 0 s – 5 s.

The trajectories of the hinge and ball joint of the first five seconds of simulation time are shown in fig. 3.1 as well as the pendulum positions and elastic deformations in 1 s intervals. A double pendulum is a chaotic system, which can be seen at the trajectory of the second pendulum, following no predictable path. The circular path of the ball joint indicate that no relevant constraint drift is present. No visible deflection at the ball joint is present either.
In fig. 3.2 and fig. 3.3 the corresponding quaternion velocities are shown. Since the system is located in the $x-z$ plane, there is obviously no rotation around any axes other than the y-axis. When recalling eq. (2.17) and eq. (2.18), it is obvious that $\dot{q}_1 = \dot{q}_3 = 0$ during the simulation.
Quaternion derivatives feature an elegant way to state the EOM of flexible bodies and Jacobian calculation is simplified significantly. A drawback of this method

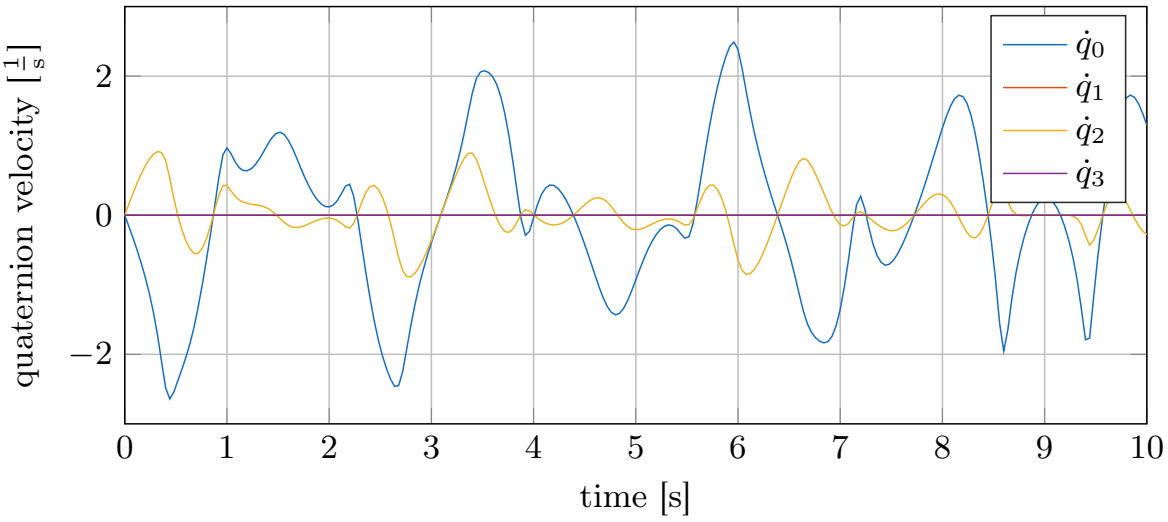


Figure 3.2: Quaternion velocity of the first beam.

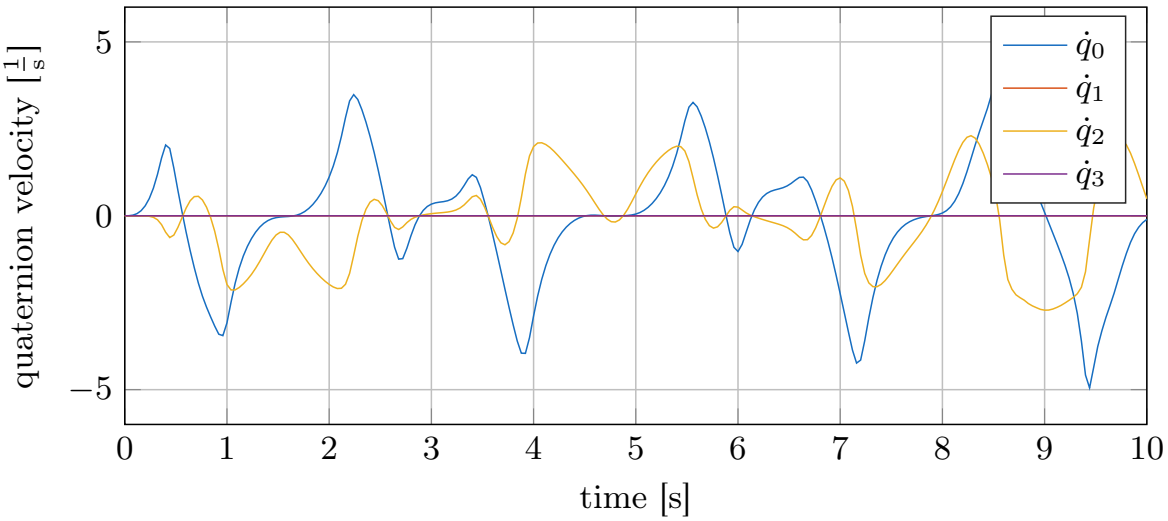


Figure 3.3: Quaternion velocity of the second beam.

is the higher system dimension and the lack of interpretability for the quaternion derivatives. Therefore in the following chapters this approach is not used for RT applications.

Vehicle Dynamics with Structural Flexibility

The field of vehicle dynamics covers the spatial motion of a vehicle on a road or off-road surface. In addition, it may as well include the motion and behavior of occupants and the vehicle's impact on road, air and other vehicles. If only a certain area of vehicle motion is of interest, reduced models can be used to gain sufficient information. Simple vehicle dynamics investigations can be done by using only vertical or lateral models as shown in figure 4.1.

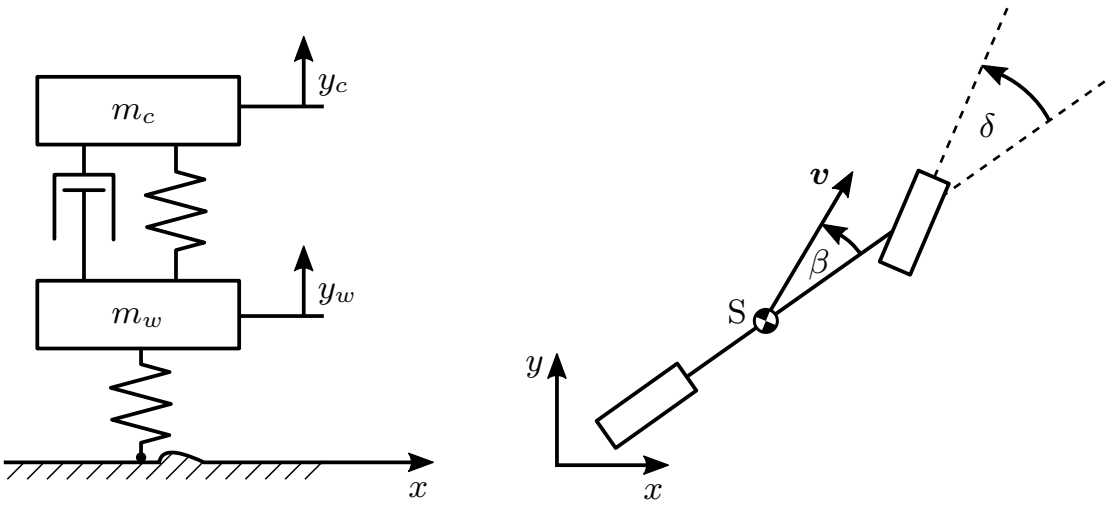


Figure 4.1: Quarter car model (left) and single-track model (right).

A quarter car model only has two DOF in vertical direction for the car body motion y_c and the wheel motion y_w. It is sufficient for basic analysis of the vertical behavior of a car body and can be used for a first estimation of suspension setup parameters and comfort investigations. The single-track model [43] allows basic investigations of steady cornering and can, for example, be used to investigate the understeering and oversteering behavior of the vehicle or for simple control algorithms. For vertical models structural flexibility can be introduced with additional sprung masses, e.g. to analyze the vertical vibration behavior of an internal combustion engine.
However, although such simple models allow basic vehicle dynamics investigations and are of course RT-capable, their value for detailed description of vehicles in versatile environments is limited. Two-track models allow the simulation of all common vehicle conditions and the calculation of vehicle rotations around all

three axes, called pitch, roll and yaw. Figure 4.2 shows the most basic two-track model with a linear suspension setup, allowing the suspension to move along a linear axis relative to the car body. Considering the car body as a free body in space with 6 DOF, having one translational DOF at each suspension and one rotational DOF at each wheel, this vehicle model yields 14 DOF in total. The connection to the environment is then realized with generally nonlinear force elements at the wheels, representing the tire forces.

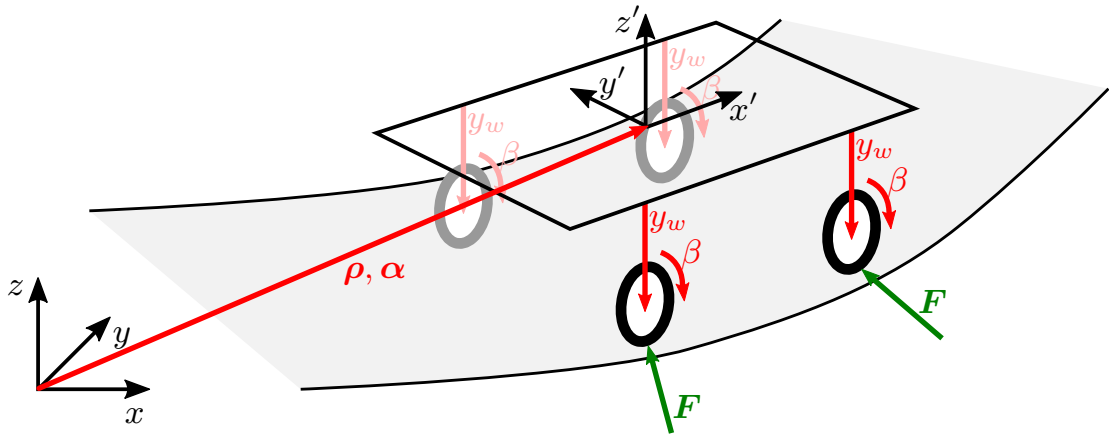


Figure 4.2: Two-track model without suspension kinematics.

Two-track models with simplified suspension kinematics have been proven to be suitable for RT simulation as well [30]. Advantages of such models are the absence of kinematic loops and hence the ability to use ODE solvers for numerical solution. To some extend these models can be used to simulate structural flexibility within the car body or other vehicle components like suspension parts. The following sections cover the integration of structural flexibility in vehicle dynamics.

4.1 Aspects on Vehicle Dynamics with Structural Flexibility

Several parts of a state-of-the-art road vehicle are prone to elastic deformation that may influence the handling, comfort or durability of the vehicle. While durability and some aspects of comfort like acoustic effects are hardly relevant for real-time applications, structural flexibility may be relevant when influencing the dynamic behavior of the vehicle [34]. The lower eigenfrequencies of the structural assembly of a modern passenger car with significant influence on vehicle dynamics are the torsional and bending mode. These typically lie in the range of 20-40 Hz, depending on the body design of the car. Figure 4.3 shows the torsional eigenmode of a passenger car which e.g. may get induced by driving on a cobblestone road.
Flexibility of other vehicle components may as well have a great influence on the dynamic vehicle behavior and hence modeled as a flexible body. Lightweight

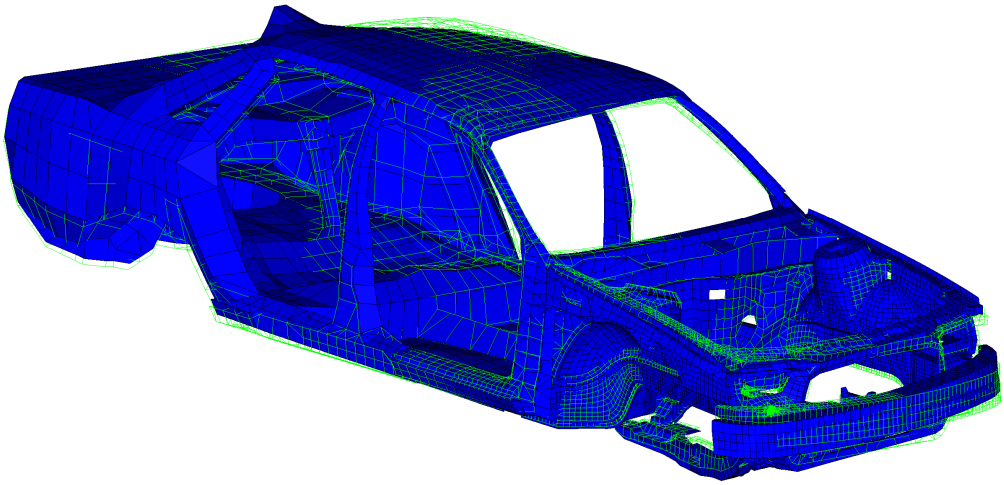


Figure 4.3: Torsional eigenmode of a Ford Taurus from [33] at 24.5 Hz.

design is advantageous especially for suspension components since it reduces the unsprung masses of the vehicle. Thus for most low-priced vehicles the front suspension wishbone is designed as a sheet-metal part that may undergo significant elastic deformation [15]. Other suspension setups like torsion-beam suspensions or leaf-spring suspensions can be simulated as flexible bodies as well [20, 64]. Half-rigid suspension setups like the torsion-beam axle are intentionally designed to behave elastically in driving maneuvers. These designs are of particular interest for flexible multibody simulations.

4.2 Current Approaches for Flexible Bodies in Vehicle Simulations

In section 2.2 several methods to solve the EOM in RT are presented. In general, all these methods can be used to simulate vehicle dynamics models with structural flexibility in RT. However, structure of the vehicle setup, field of application and limited computational power have a large influence on the model topology.

Finite Segment Models

When only few deformation modes are of interest, a rigid body model may be extended as a finite segment model. For example, the car body of the two-track model in figure 4.2 can be divided into two rigid bodies connected by a rotational joint and a torsional spring-damper setup. This already gives a first estimate of the chassis torsional stiffness or bending stiffness and may be sufficient for most RT applications. A major drawback is the need for measurement data to parametrize the torsional spring-damper setup [48].

Beam Models

To keep the number of DOF low while still covering a large amount of the deformation modes, the car body or other parts like a stabilizer bar can be modeled with few beam elements. Figure 4.4 shows a basic beam model covering the main load paths within a passenger car. Since the load paths shown provide the majority of structural stiffness of a self-supporting car body, the deformation of the beam model will behave similar to the car body. Like the finite segment model it is necessary for the beam model to parametrize the beams with estimated geometric data and material properties. Sufficient parametrization data may be gathered experimentally by component or vehicle tests. Alternatively, FE models of full vehicles or components can be used to estimate necessary data.

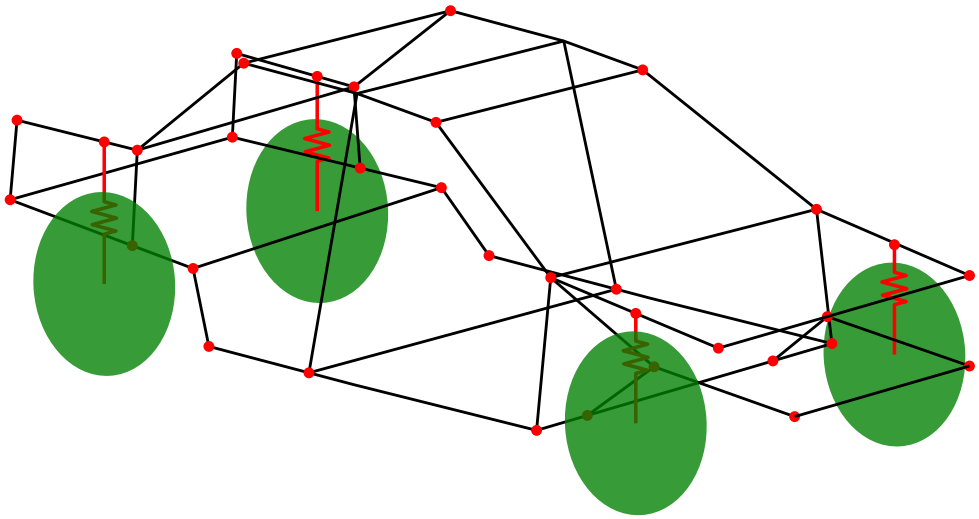


Figure 4.4: Beam model of a passenger car.

Beam models may as well be used for modeling suspension parts. For example, in [32] a stabilizer bar is discretized with beam elements. For this example the stabilizer bar beam elements still yield 120 elastic DOF, which has to be reduced significantly to be employed in RT applications.

Finite Element Models

Detailed FE models can provide extensive information on the elastic deformation of a car body. These vehicle models can be very detailed with several thousand to millions of elements. This allows not only the simulation of structural deformation occurring during driving maneuvers but also the effect on non-structural parts like doors or interior. Detailed information of the loads on body parts can be retrieved and used for stability and durability analyses. Since the number of elements used to model a vehicle massively exceeds the number of DOF that

can be simulated in RT, model order reduction methods have to be employed, as shown in section 2.5.
While geometric information of a car body can be modeled and parametrized easily, the estimation of structural damping occurring in a car body structure is very difficult. Most chassis parts are assembled with spot welds, laser welds, punch riveting or adhesive bonding methods. This introduces mostly nonlinear structural damping which has to be estimated with modal damping in case of model order reduction [35]. Compared to finite segment and beam models a FE model allows vehicle dynamics simulations in relatively early design stages of a vehicle. Since shape and material information of the chassis is known before structural parameters can be identified from real vehicle experiments, virtual prototypes that are for instance used for crash worthiness certifications can be employed to derive a suitable FE model.

4.3 Setup of a Vehicle Model With Structural Flexibility

A rigid body model is the basis for considering structural flexibilities in vehicle dynamics simulations. It can be build up with the methods as described in section 2.3 for reference purposes and as a baseline for including flexible bodies. Besides the flexible bodies that are to be included in the vehicle dynamics simulation, methods to model the suspension and tire behavior have to be chosen according to the demanded accuracy and available calculation power. The following sections give a brief overview on the modeling options as well as on RT capable road and tire models.

4.3.1 Suspension Setup and Kinematics

The purpose of a road vehicle suspension is the decoupling of the road surface from the chassis while ensuring permanent contact between tire and road. This allows the application of longitudinal and lateral forces onto the vehicle. This is realized by allowing the wheel to travel vertically relative to the car body. For steering axes the suspension also allows the rotation of the wheels along a mostly vertical axis. Smaller vehicles like go-karts and trailers can be built without a suspension. In this case tire force distribution is maintained only by tire deflection and elastic deformation of the chassis, which can be investigated with FMBS as well [19]. Railed vehicles pursue a similar setup to decouple the chassis from the track, but since the contact between wheelset and rail is very stiff, a secondary spring between wheelset and undercarriage yields similar suspension properties like road vehicles [41].
Most modern cars consist of a independent suspension setup, which allows each wheel of the car to travel vertically without influencing the movement of other wheels. Rigid and semi-rigid suspension setups play a minor role in road vehicle suspensions with the exception of torsion-beam axles for rear suspensions of compact cars [56]. McPherson type suspensions are the most common setups

among smaller and lightweight cars, while heavier cars and performance cars usually feature a double wishbone setup, see figure 4.5.

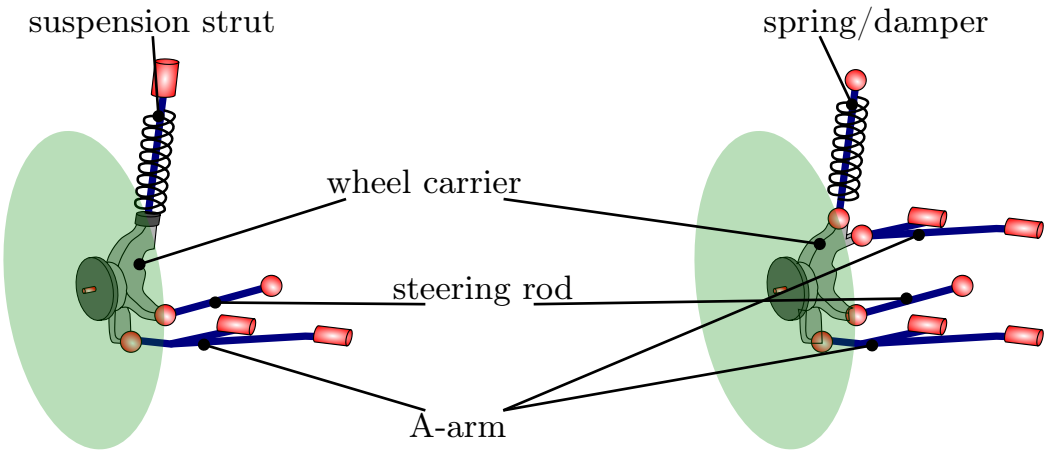


Figure 4.5: McPherson type suspension setup (left), double wishbone (right).

Both suspension setups consist of a lower A-arm that rotates around an axis fixed at the car body. While a double wishbone suspension features a second A-arm at the upper part of the suspension, a McPherson type suspension features a suspension strut instead. This suspension strut allows vertical travel along an axis fixed at the wheel carrier and pivoted at the suspension dome of the car body. The suspension strut is usually combined with a spring-damper combination. Rotational movement of the wheel carrier is controlled by a steering rod. Instead of an A-arm both suspenison setups may feature two connection rods.
Since the weight of wheel and wheel carrier is usually much larger than the weight of the connecting elements, their mass can be neglected or distributed among the surrounding components. This allows modeling a suspension with just two bodies – wheel and wheel carrier. The connecting elements can be modeled either as rigid and hence algebraic links or as rather stiff elastic components, which has been proven to be a suitable option for RT capable rigid MBS [47]. While algebraic equations result in DAE, stiff elastic components yield a set of stiff ODE and require small time steps for time integration.
The suspension setups shown above result in kinematic loops since the wheel carrier is attached to the car body at several points. For rigid chassis and suspension components the algebraic equations can be resolved analytically [44, 51]. However, when modeling the car body as a flexible body, the algebraic equations cannot be resolved since the position coordinates of the attachment points can move relative to the chassis reference frame. This requires the vehicle model to be set up as a system of DAE when considering car body flexibility.
For a limited range of applications it may be sufficient to replace the complex suspension kinematics with a simplified setup consisting of only one translational joint at the suspension dome [52]. The suspension dome supports the vertical load of the vehicle since it is the application point of the spring and damper

forces. Some driving maneuvers show dominant vertical excitation and the lateral and longitudinal forces from the wheel carrier onto the car body may be neglected. Then these can be replaced by relatively small guidance forces and torques at the suspension dome.

4.3.2 A Real-Time Capable Tire and Road Model

Under normal driving conditions a road vehicle is connected to the environment only with tire and drag forces. Hence the tire model combined with a suitable road representation is a crucial part of vehicle dynamics simulations. A standard road tire is built up as a pressurized tubeless rubber tire with a steel-reinforced rubber belt. The road contact is realized with various rubber compounds depending on the field of use. For tires used on public roads, the tread usually contains studs to improve behavior in wet conditions. Tire modeling is depending on the field of application and desired accuracy, which can vary widely depending on modeling of carcass, contact zone and contact forces. Figure 4.6 shows three common tire modeling approaches used in vehicle dynamics simulations.

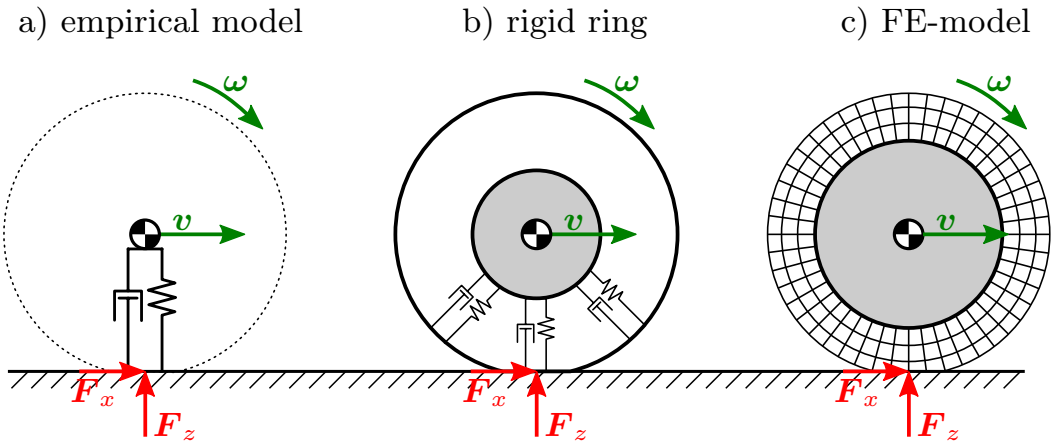


Figure 4.6: Common tire modeling approaches [21, 40].

The empirical model, see figure 4.6 a), is based only on parametrized load curves with kinematic values of the wheel as input quantities. In contrast, the rigid ring model in figure 4.6 b) considers the tread dynamic behavior with an extra body. The rigid ring may be replaced by a deformable ring to consider deformation of the loaded tire as well. These models contain a rather low number of DOF and hence are usable for RT applications. Finite element-based models, figure 4.6 c), follow a continuum-based approach and allow detailed investigation even of high-frequency behavior and highly transient maneuvers [68]. However, tire modeling with finite elements show a high computational load and usually do not allow RT capable simulations [21].
For steady-state applications and application with low accuracy requirements the dynamics in vertical direction can be modeled with a linear spring-damper

combination. Thereby the force can be calculated with respect to a single contact point or, depending on the road surface properties, a contact surface can be defined, the contact force then is distributed on this surface.

Calculation of Longitudinal and Lateral Tire Forces

Lateral and longitudinal forces are generated when the wheel moves relative to the road surface. While vertical forces can be modeled simply with a spring-damper combination, significant horizontal forces occur only if there exists a velocity difference. To understand the generation of longitudinal and lateral forces and the meaning of intermediate kinematic quantities, a brush model can be employed [40, 41], see figure 4.7.

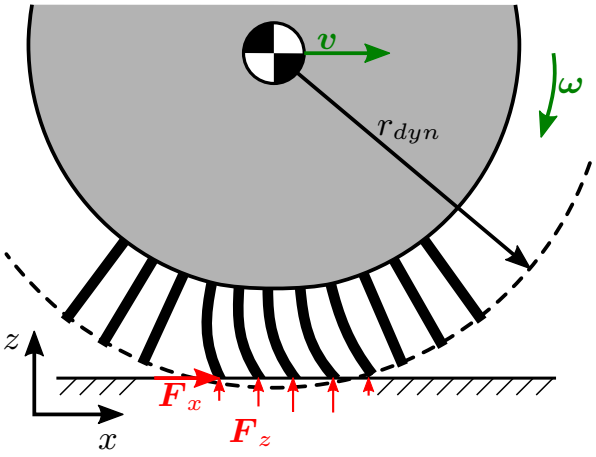


Figure 4.7: Brush model.

Longitudinal slip:

$$\lambda_x = \frac{v_{sx}}{v_x} \approx \frac{v_x - \omega_y r_{dyn}}{v_x} \tag{4.1}$$

Lateral slip:

$$\tan(\lambda_y) = \frac{v_{sy}}{v_x} \approx \frac{v_y}{v_x} \tag{4.2}$$

with $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$

$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}$

With the rather soft carcass the contact area deforms slightly under horizontal load of the tire, similar to a brush pressed on a surface and pushed forward or sideways. The deformed bristles of the brush in contact with the surface generate a horizontal friction force which corresponds to the lateral and longitudinal tire forces. Since the bristles deform under load, an idealized contact point of the wheel with the road surface moves relative to the road surface. The relative motion is called longitudinal and lateral slip and can be calculated from the kinematic quantities of the wheel and the dynamic wheel diameter r_{dyn} with $r_{dyn} = \frac{v_x}{\omega_y}$ for a wheel at purely kinematic rolling. Equation (4.1) and eq. (4.2) show the most common definition of slip for a moving wheel. Thereby longitudinal and lateral slip speed v_{sx}, v_{sy} represent the horizontal velocity difference of the idealized contact point which can be approximated by the kinematic properties of the wheel for most applications [40]. Other slip definitions exist especially for low longitudinal velocities ($v_x \to 0$) and more accurate contact zone definitions.
Based on longitudinal and lateral slip the forces occurring at the contact point can be calculated. Since the force generation is highly nonlinear, they are usually

based on an empiric model fitted with measurement data. Among the empiric models the so-called Magic Formula model [40] is an efficient and accurate model of tire forces often used for RT applications.

The Magic Formula Model

The Magic Formula tire model is an empiric tire model based on trigonometric functions, allowing the calculation of tire forces under a broad variety of operation conditions. The basic equation to describe the relationship between slip and force is given as

$$F(\lambda) = D \sin\left(C \arctan\left(B\lambda - E\left(B\lambda - \arctan\left(B\lambda\right)\right)\right)\right) \ . \tag{4.3}$$

The parameters $B - E$ have to be determined from extensive real-world or numerical experiments [50]. The main advantage of the Magic Formula (MF)-tire model is the relatively low number of parameters to describe a realistic load curve. With exemplary parameters from [40] eq. (4.3) yields the tire force versus slip curve shown in figure 4.8. It can be seen that for low slip conditions ($\lambda < 0.1$) there is an almost linear relationship between slip and tire force, describing the behavior under regular driving conditions. The force curve reaches a maximum at $\lambda \approx 0.15$, comparable with the static friction coefficient, and decreases for high slip values, which corresponds to a spinning or sliding tire.

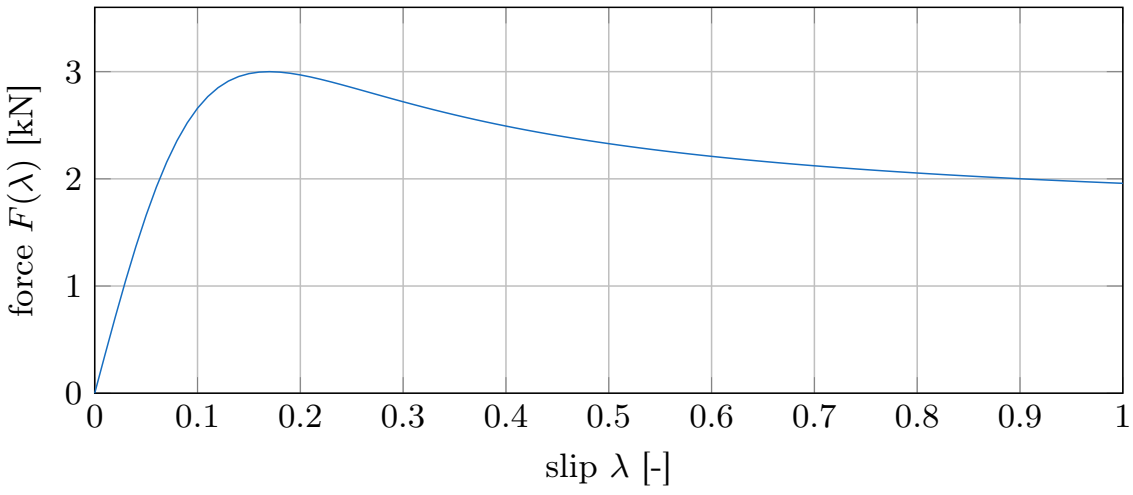


Figure 4.8: MF-tire force versus slip curve.

The basic equation (4.3) can be extended with additional parameters to describe the influence of tire pressure, rapidly changing vertical tire forces and road surface conditions. Additional equations are defined for combined slip conditions and torque load occurring at the tire contact point in [40].

Road Surface Model

The MF tire model describes the tire forces with respect to longitudinal and lateral slip. While the tire contact point, and hence the slip quantities, can be calculated easily for a horizontal surface, a road model is necessary for uneven surfaces. Detailed road modeling can become complex for many driving conditions like a wet road surface or soft soil conditions. For an undeformable, dry road the surface can be described as a geometric surface in 3D. If only small road sections have to be defined, the surface can be modeled in the global coordinate system as a x, y grid. However, for longer road sections it is more useful to introduce an intermediate coordinate system that is related to the current orientation of the road center line. In [66] this approach is realized as an open-source plug-in that can be used for RT applications, called Open Curved Regular Grid (OpenCRG). The OpenCRG road model first describes a reference line relative to the global coordinate system. Relative to the u, v coordinate system attached to the reference line the road surface is defined. Figure 4.9 shows an exemplary section of a cobblestone road and the reference line used to define the surface.

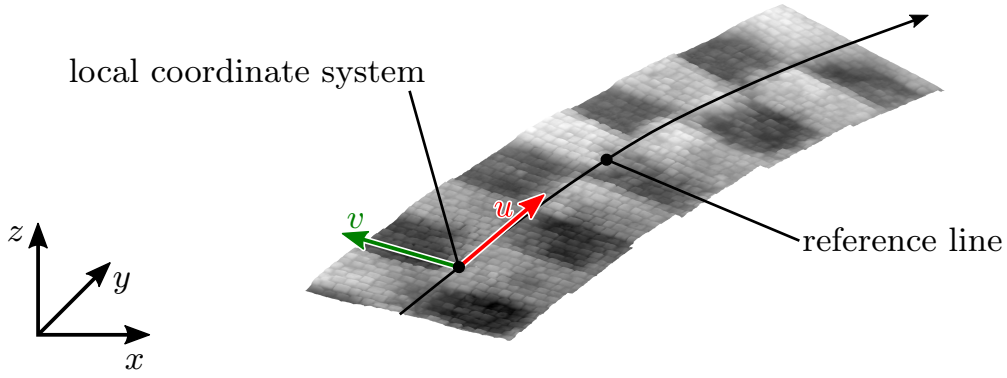


Figure 4.9: Cobblestone road with curved reference line [66].

4.4 Limitations of Flexible Bodies in Real-Time Environments

As described in section 4.2, several different approaches exist to include body flexibility in vehicle dynamics simulations. While in principle all methods shown are suitable for RT simulations by employing model-order reduction methods, the resulting number of elastic coordinates still is a limiting factor due to the calculation power available. In [53, 54] it is shown that for a full vehicle dynamics simulation it is possible to include more than 50 additional elastic DOF while still preserving RT capability.
Even if the computational power is sufficient, elastic bodies still may be unsuitable for RT simulation. In [52] explicit solvers, as shown in section 2.2.2, are

used to perform the time integration of a FMBS formulated in generalized coordinates. An eigenvalue analysis of the linearized system yields eigenfrequencies close to the stability limit of the solver even for low step sizes. Hence the elastic modes retained in the flexible bodies and step size of explicit solvers have to be chosen appropriately to ensure numerical stability for all expected driving maneuvers.

The accurate modeling of independent suspension setups requires the attachment of suspension components at several interface points of the vehicle body, yielding kinematic loops. Although there exist approaches for resolving kinematic loops for some independent suspension setups [44], these approaches are valid only for interface points fixed to a rigid body chassis. Introducing a flexible car body allows the interface points to move relative to each other, making it impossible to resolve kinematic loops analytically. This limits the modeling options to solve redundant coordinates with RT capable solvers as shown in section 2.2.3. Compared to the formulation in generalized coordinates this yields an increasing system size and hence consuming large amounts of calculation power.

While implicit solvers allow DAEs to be solved in RT, the step sizes used for implicit methods are generally quite large. This can yield issues with the calculation of applied forces, since especially for suspension components the applied forces are very stiff. A common issue is numerical instability, see section 2.2.2. Another phenomenon arising with stiff forces and large numerical step sizes is the occurrence of unnatural vibrations at discontinuities, for example a tire at lift-off or dry friction. A workaround for this problem are so-called global derivatives as proposed in [2].

4.5 Finite Element Models of Car Bodies

Finite element models of vehicle bodies exist for various purposes due to their versatile options for applications. Most models are used during the development of a vehicle to validate strength, acoustic and NVH behavior and are not available for research purposes. Besides these models, several ones have been developed to investigate the crashworthiness of vehicles under various crash scenarios [11]. These models are based on real vehicles that are scanned, digitized and meshed for the simulation of various crash configurations. Hence, the models feature all structural components of the vehicle and also non-structural parts that may be relevant for occupant or pedestrian safety. Figure 4.10 shows the full vehicle model of a 2012 Toyota Camry used for the simulations in this work.

The model not only includes the structural components but also engine, interior, doors and suspension components, resulting in more than 2 million elements with an average element size of 6–8 mm. The structural components are meshed with shell elements and connected with spotweld elements at the exact locations of the real vehicle.

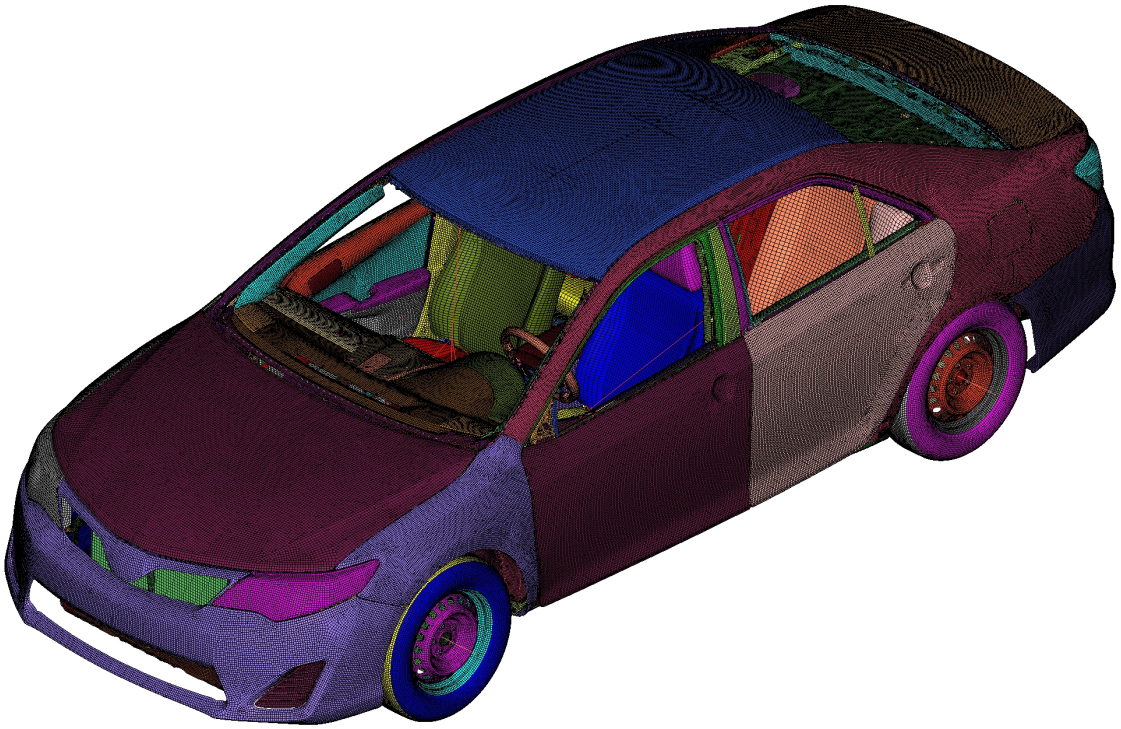


Figure 4.10: Full vehicle model, 2012 Toyota Camry [42], windows not shown.

4.5.1 Model Preparation for Vehicle Dynamics Simulations

In order to prepare this model for vehicle dynamics simulations, several preparations are performed. All non-structural components are removed so that only the body in white is left. The removed components include

- all non-structural body panels. Bumpers, hood, trunk, doors. Interior parts like dashboard, seats.
- engine and transmission parts. The engine is mounted on a subframe with soft bearings to dampen the engine vibrations. The engine can be replaced by a rigid body connected to the car body with spring-damper elements.
- suspension parts. The suspension is removed and replaced by rigid body elements and algebraic constraints in the flexible MBS.

After removal of all unnecessary parts the FE model consist of 1.02 million shell elements. Individual parts are connected only by algebraic constraints between FE nodes, also called Rigid Body Elements (RBEs). Figure 4.11 shows the finished model with a detailed view of connections between the sheet metal parts in which the spotwelds are replaced by RBEs. Windshield and rear windows are retained in the model since the glass is firmly attached to the car body structure and hence also affects the structural flexibility. In crash worthiness simulations the reference frame of the vehicle is located on road level at the foremost point

of the car. The FE model is moved so that the reference frame is located at the car body CoG, representing a Buckens frame [57].

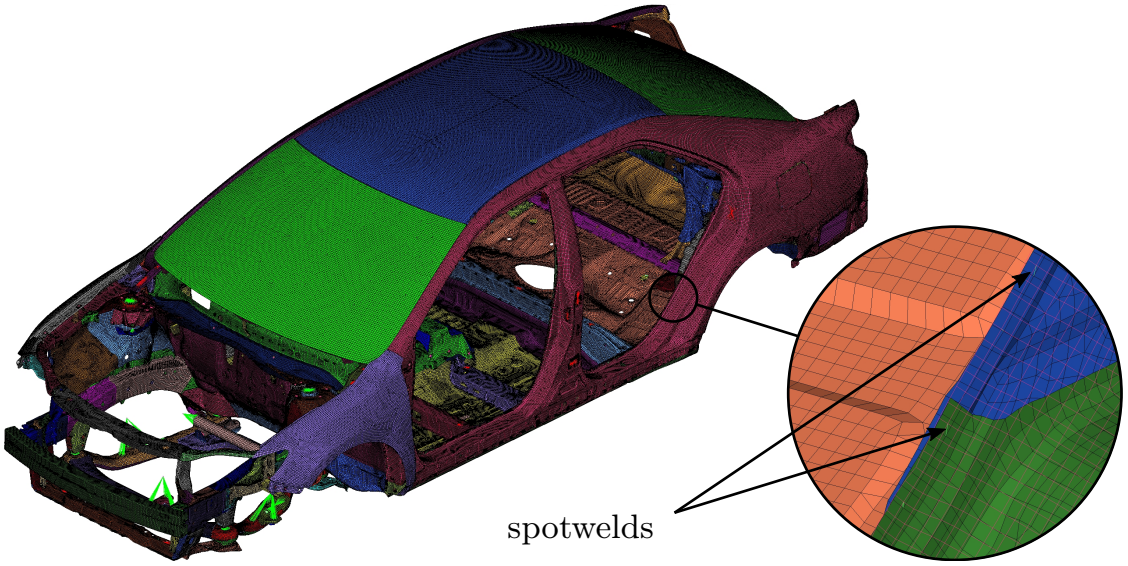


Figure 4.11: Reduced FE model of the 2012 Toyota Camry.

The suspension of a Toyota Camry consists of four McPherson type setups, attached to the vehicle at four interface points each. To improve passenger comfort, the connecting rods and arms are attached with rubber bearings, allowing little travel between adjacent bodies. The rubber elements are replaced with RBE interfaces that define a center interface node, connected to the surrounding structure with a spider-like connection, see Figure 4.12.

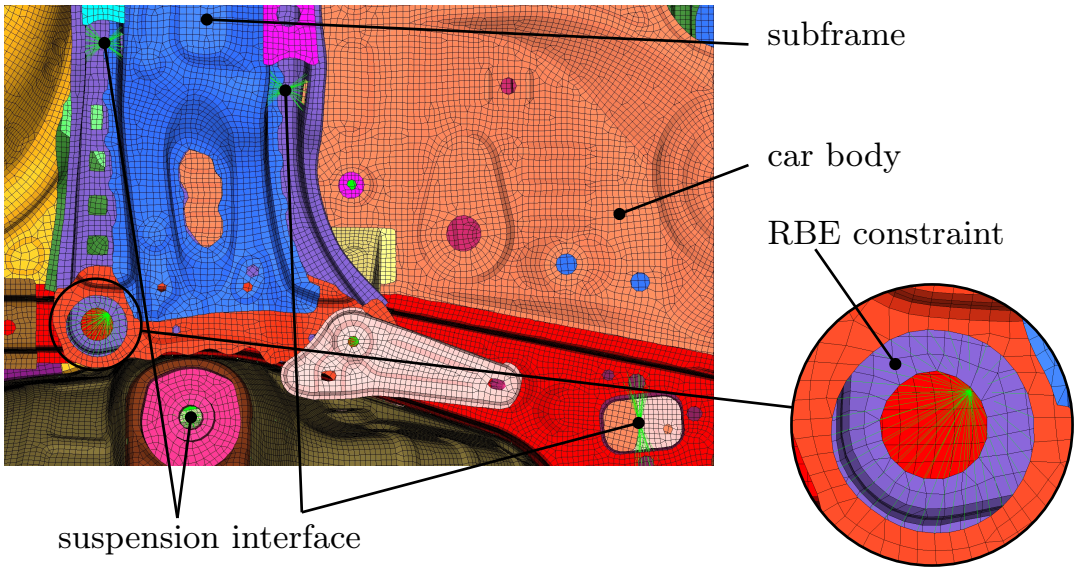


Figure 4.12: Rear left subframe and suspension interface.

Front and rear suspension are partially connected to a subframe that increases the stability of the suspension interface and simplifies manufacturing since the suspension can be attached to the car body as one assembly. The engine is supported at the front subframe as well. Subframes and vehicle body are connected with RBEs as well to create a single body that can be used for the vehicle dynamics simulations.

4.5.2 Model Order Reduction of a Car Body

The model order reduction (MOR) process in this work is based on a 2012 Toyota Camry model for crash simulation purposes. The model is available for the FE solver *LS-Dyna*, which is commonly used for transient crash simulations. For editing and conversion of a *LS-Dyna* model to the FE solver *ANSYS* the preprocessor *HyperMesh* is used in this work. After the FE geometry of the vehicle body is transferred from the crash simulation model to a model suitable for structural flexibility investigation, the MOR process described in section 2.5 can be conducted and is shown in figure 4.13.

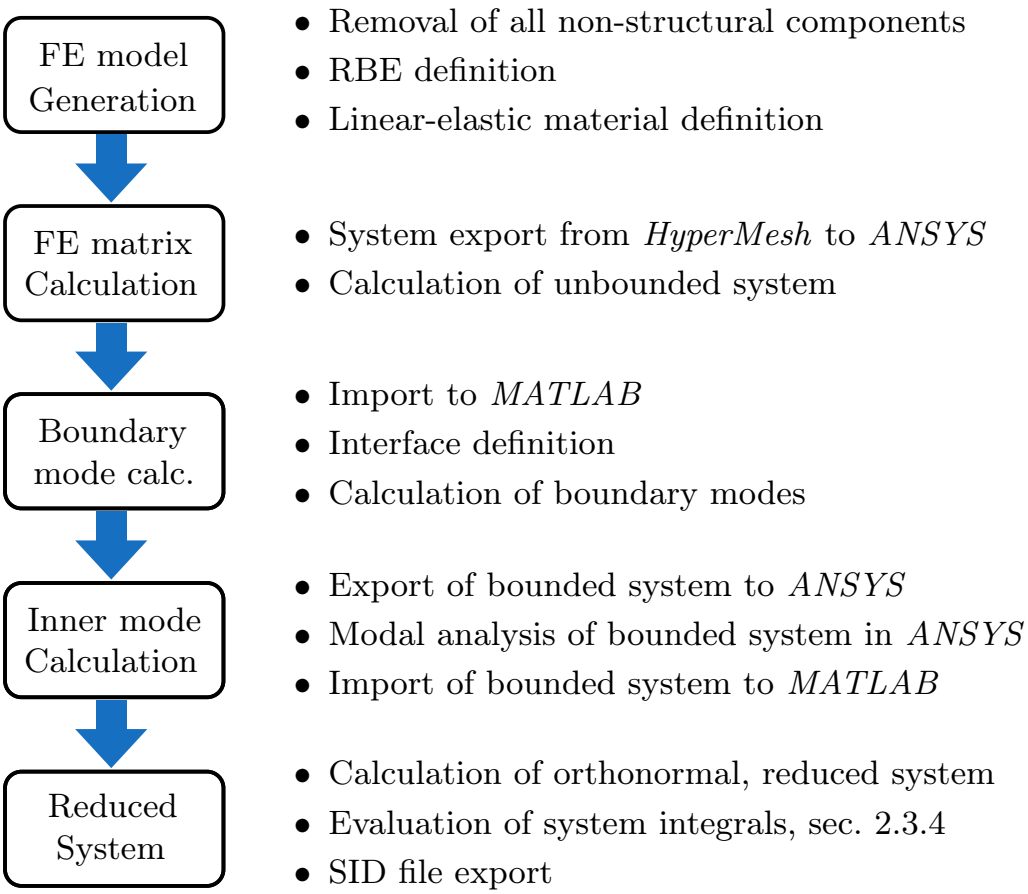


Figure 4.13: Process of SID data generation for a large vehicle model.

This process involves several consecutive steps that usually can be executed within a numerical environment like *MATLAB*. The converted FE model consists

of 6.24 million DOF, due to the large-scale FE model the model reduction process is performed partially in *MATLAB* and *ANSYS* to allow the process to be executed on a desktop workstation.
Definition of the boundary modes and selection of the inner modes largely influences the system behavior in flexible multibody simulations. The vehicle model requires interface nodes at all suspension attachment points and engine attachment points. However, using all interface nodes as boundary nodes would require a large number of elastic DOF and limit the number of inner DOF to be retained in the reduced system. Therefore, to circumvent an unnecessary large system, only the interface nodes for major force application points are used as boundary nodes in this process. The suspension domes support the car body weight and act as the force application points of the suspension springs. Hence the suspension domes are chosen as boundary nodes, the remaining interface nodes are not used for the bounded system. Overall the reduced model consists of 46 shape modes, 12 boundary modes and 34 inner modes. The 12 boundary modes consist of 3 translational DOF at each suspension dome. Six rigid body modes are removed after the MOR process and the final model consists of 40 elastic coordinates.

Real-Time Vehicle Dynamics Simulation With Structural Flexibility

The methods and numerical solvers presented in chapter 2 are tested and validated with several numerical examples. The vehicle model assembled based on the data available from [42] is transferred into a RT-capable vehicle dynamics simulation and validated with a standard driving maneuver using the commercial MBS software *ADAMS*. The procedure pursued to gain the vehicle model is shown in figure 5.1. Besides the extraction of the flexible body it is important to gain geometric and inertial data of all required rigid bodies as well.

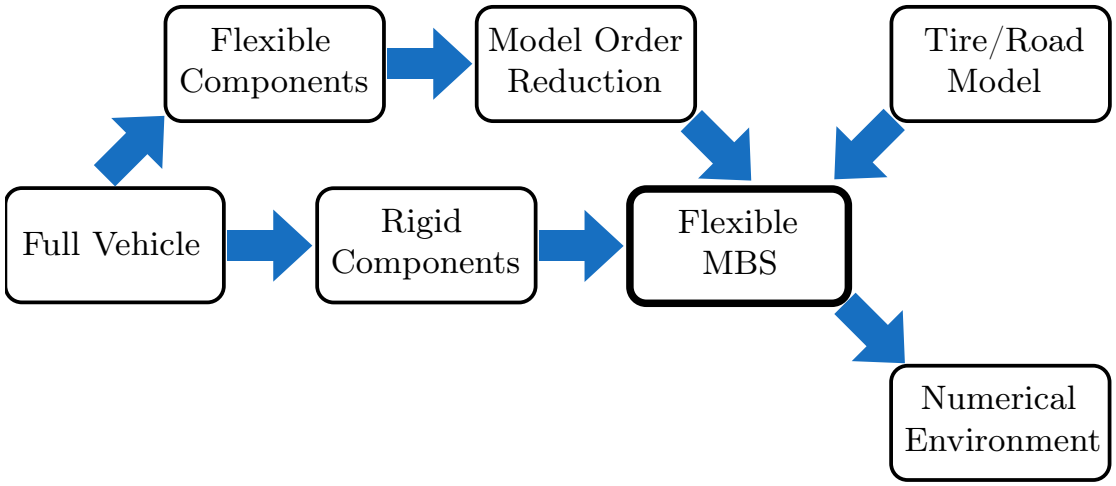


Figure 5.1: Toolchain for flexible vehicle dynamics simulation.

The numerical environment chosen for the model setup is *MATLAB/Simulink* [65], which provides a broad choice of built-in numerical integrators and the option to generate RT-capable C-code from the generated models. Several target platforms are supported to use the models in RT environments, which may be a field of use for future work.
In section 5.2 the commercial MBS software used for validation is *ADAMS* [36], a common choice for MBS simulations. The software formulates the EOM in cartesian coordinates and introduces constraints as algebraic equations. The resulting set of DAE can be integrated on index-3, index-2 and index-1 level with built-in integrators. The MF-tire model and several road models including the OpenCRG model are included in this software, which allows an easy comparison of simulation results.

Several models are tested and shown in this chapter. The methods on index-2 and index-1 formulation of constraint equations shown in chapter 2 is realized in two models that only vary in the constraint formulation. The linearly implicit Euler scheme is implemented for the index-2 model and shown in section 5.1.2. The built-in *ODE4* solver of *Simulink* is used for the index-1 model in section 5.1.3. To demonstrate the differences between rigid and flexible MBS the simulations are compared with identical simulation models without flexible bodies in section 5.3.
Generalized coordinates are suitable for heavily constrained systems and tree-like structures, but not usable for kinematic loops like they occur in suspension setups. In contrast to the suspension kinematics that are usually connected with connection rods, wheel carrier and wheel are constrained by a rotational joint, removing five DOF from the system. By creating a generalized subsystem that combines wheel carrier and wheel to one subsystem with 7 DOF and a conventional DAE formulation for car body and suspension constraints, the system dimension can be reduced significantly by 24 coordinates and 20 constraint equations. Such a model is shown in section 5.1.4 and compared to the other models.

5.1 Simulation Setup

This section first describes the vehicle model used for numerical examples and model validation in this thesis, afterwards the setup of the numerical index-2 model is explained. The realization of an index-1 formulation is outlined as well and the vehicle model with generalized subsystems is shown.

5.1.1 Mechanical Properties of the 2012 Toyota Camry Model

From the removed non-structural parts of the FE model the mechanical properties are retrieved to match later the MBS with the original model. The vehicle is modeled with 10 bodies, which are the flexible car body, four wheel carriers and wheels and the engine. An overview of the vehicle is shown in figure 5.2 including all constraints and applied forces. The inertial properties of the bodies are shown in table 5.1.
As shown in section 4.5.1, the FE-model for crash simulation purposes is transferred to an *ANSYS* model and reduced to an elastic body with 40 elastic coordinates. This results in the flexible car body which is defined in an SID structure, see section 2.3.4. It consists 12 boundary modes (3 translational boundary modes at each suspension dome) and 28 inner modes. The interface to the other bodies of the MBS is defined by 20 interface nodes, four for each suspension and four for the engine attachment points.
All bodies are constrained by a total of 21 constraints, see figure 5.2. Three connection rods and one McPherson constraint are used to attach each wheel carrier

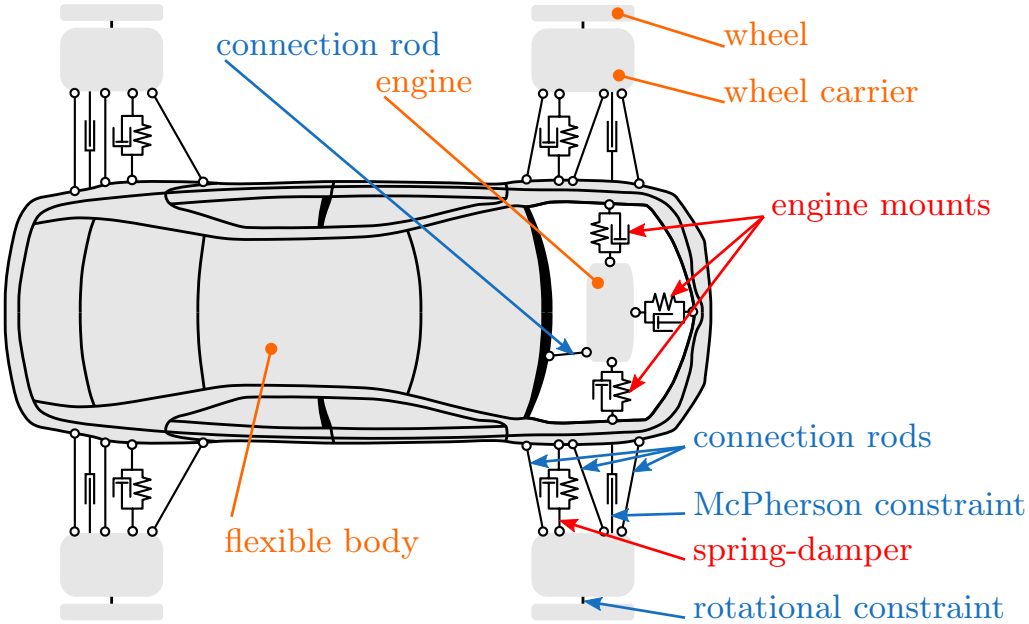


Figure 5.2: Structure of the 2012 Toyota Camry.

to the car body. The term McPherson constraint is used for a translational constraint representing the suspension strut kinematics of a McPherson strut. It limits the movement of the suspension dome interface point at the car body to an axis defined in a local coordinate system attached to the wheel carrier and oriented along the suspension strut axis. Hence it is partly a ball joint constraint defined in the suspension strut axis local coordinate system. The A-arms at the front suspensions allow the rotation of the respective lower wheel carrier hinge around an axis defined by the two attachment points of the car body attachment point. Instead of defining a new constraint, the A-arm is replaced by two connection rod constraints attached to the same node at the wheel carrier, creating the same circular rotation like the A-arm.

A rotational constraint is used to attach each wheel to the wheel carrier. The engine resides on the front subframe of the vehicle with three engine mounts. An additional torque support bar on the top right corner of the engine is modeled with a connection rod constraint as well.

All constraint definitions with the respective interface points and the number of constraint equations are listed in table A.2. To further visualize the model setup, the FE model design and the MBS implementation is shown for the front suspension in figure A.1 and figure A.2, for the rear suspension in figure A.3 and figure A.4 and for the engine in figure A.5.

Suspension strut forces are modeled as a linear spring-damper element acting on the same points like the McPherson constraints. Suspension end-stops are neglected in this model as well as nonlinear spring and damper behavior, but may be included for simulations with large suspension travel and high dynamic requirements. Engine mounts are represented by three-dimensional linear spring-

Table 5.1: Mass and CoG location.

Part	Mass	CoG location
Car body	1006.5 kg	$\begin{bmatrix} 0.000 & 0.000 & 0.602 \end{bmatrix}$ m
Wheel carrier FL	17.32 kg	$\begin{bmatrix} 1.413 & 0.765 & 0.297 \end{bmatrix}$ m
Wheel FL	39.04 kg	$\begin{bmatrix} 1.413 & 0.805 & 0.297 \end{bmatrix}$ m
Wheel carrier FR	17.32 kg	$\begin{bmatrix} 1.413 & -0.765 & 0.297 \end{bmatrix}$ m
Wheel FR	39.04 kg	$\begin{bmatrix} 1.413 & -0.805 & 0.297 \end{bmatrix}$ m
Wheel carrier RL	22.94 kg	$\begin{bmatrix} -1.377 & 0.750 & 0.299 \end{bmatrix}$ m
Wheel RL	28.35 kg	$\begin{bmatrix} -1.377 & 0.790 & 0.299 \end{bmatrix}$ m
Wheel carrier RR	22.94 kg	$\begin{bmatrix} -1.377 & -0.750 & 0.299 \end{bmatrix}$ m
Wheel RR	28.35 kg	$\begin{bmatrix} -1.377 & -0.790 & 0.299 \end{bmatrix}$ m
Engine	254.63 kg	$\begin{bmatrix} 1.558 & 0.000 & 0.460 \end{bmatrix}$ m

damper force elements with the same limitations as for the suspension forces. The force element data used for the simulations is summarized in table A.1

5.1.2 Numerical Index-2 Model in MATLAB/Simulink

MATLAB/Simulink is a block-based numerical programming environment, which means that, compared to writing executable code manually, the functionality is represented by function blocks and signals. Function blocks provide various features like mathematical calculations, introducing system states, storage or visualization. Several blocks can be grouped into subsystems, forming new blocks and allowing a structured layout. Blocks can be compared to function statements in a text-based programming language. Signal lines hence are the counterpart to variables. They can state scalar as well as multidimensional numerical values of any data type.
Simulink features some options to improve design and clarity of large models. These features are used extensively in this work to keep the model manageable and versatile.

- **Bus signals** are used to summarize signals belonging to one object, like the kinematic values of a body or force components of a tire.

- **Libraries** are subsystems that can be reused within a model, for example the calculations of a force element or a constraint.

- **Visualization** A Virtual Reality (VR) toolbox allows to visualize results during a simulation.

- **Solvers** Common one-step RK methods are already included in the environment, but are limited to ODE and index-1 DAE.

The basic layout of the *Simulink* model with index-2 formulation is shown on the top level in figure 5.3. The signal flow from left to right starts with calculation of all body-related quantities like inertial data, kinematic data of nodes and inertial and internal forces. The library Bodies outputs a bus which itself contains an extra bus for each body with the body-related quantities. For flexible bodies the SID structure described in table 2.1 is used to calculate the body-related sizes dependent on the elastic coordinates. Based on the body-related quantities all constraint equations and applied forces are calculated in the next steps. Furthermore, global matrices are assembled as a preparation for the next integration step.
MATLAB/Simulink is designed to solve ODE and index-1 DAE only with its built-in solvers. The solution of the index-2 EOM with the linearly implicit Euler scheme is implemented manually as a discrete time integration method. The time integration process starts with the solution of the linearly implicit Euler method described in section 2.2.3, eq. (2.125), which is implemented in the green I2_Update_Fully_Implicit block. Since the solution may drift off the constraint manifold on position and velocity level, so the solution is corrected by a projection step (bright blue Projection block) according to eq. (2.128) in the final blocks of the integration process. The magenta-colored Time_Step block delays the state vector by one time step and hence initializes the process for the following time step.
In figure A.6 the linearly implicit Euler method subsystem is shown, containing the semi-implicit matrix assembly in the lower part, the right side of the equation is assembled in the upper half.
Since the entries of the state vector

$$\text{states} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}$$

are dependent on each other, the simulation has to be started with consistent initial conditions. The block Algebraic_IC short-cuts the kinematics and constraint evaluation, creating a so-called algebraic loop. *Simulink* allows the solution of algebraic loops with an iterative method, hence consistent initial conditions can be retrieved with the same model that is used for time integration afterwards.
Implicit integration requires calculation of the Jacobian matrix $\boldsymbol{J}$, see eq. (2.122). Several approaches exist to calculate or approximate the Jacobian matrix, see [10, 22]. Common methods are

- Calculation of an initial Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_0, \boldsymbol{z}_0, t_0)$ that is used for the whole integration process. For systems with high nonlinearities the

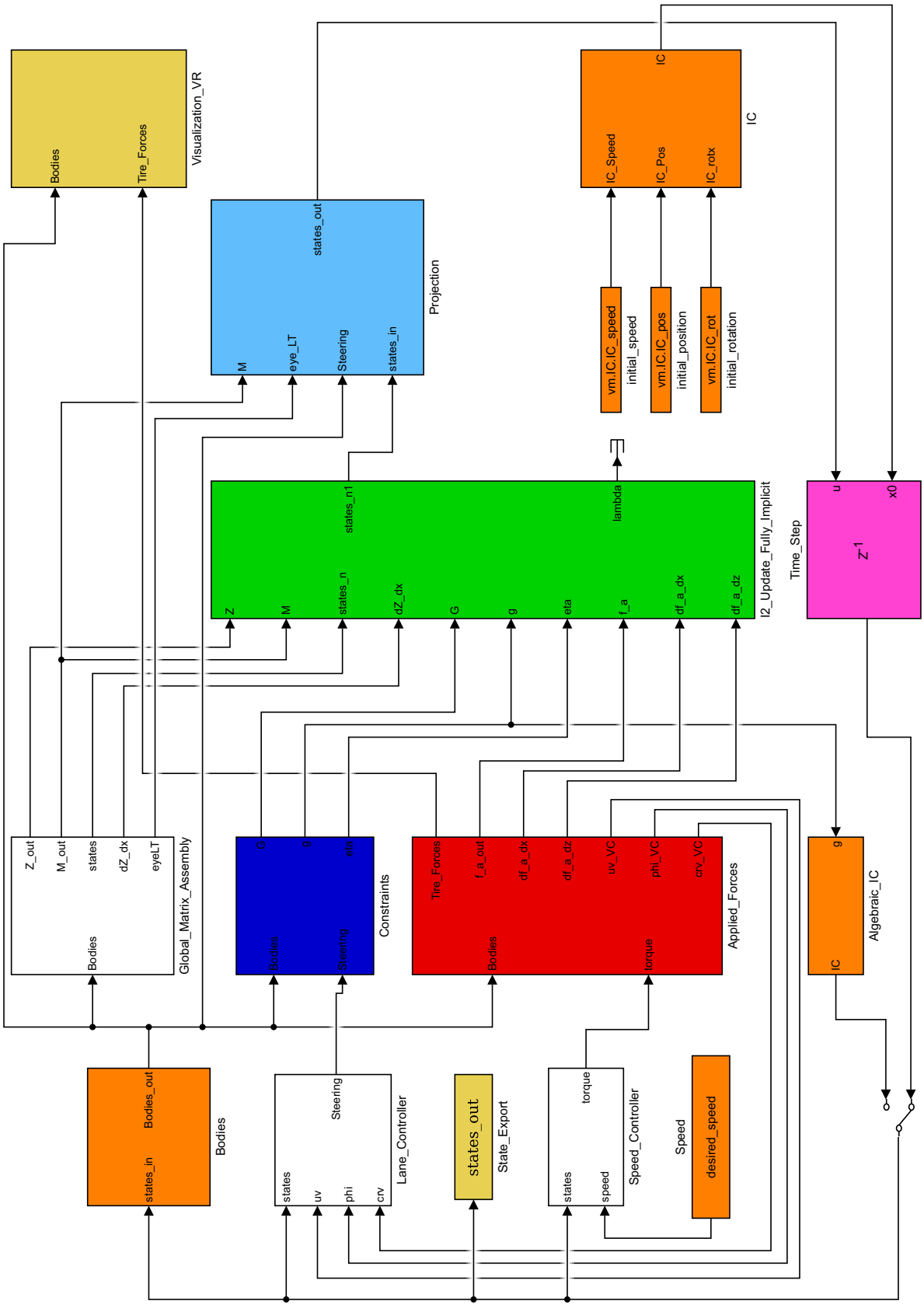


Figure 5.3: Top level layout of the index-2 *Simulink* model.

Jacobian may be updated during the integration process. However, this method is not RT capable since the update interval is not predictable.

- Approximation with finite differences. This method requires complex recalculation during the simulation process, especially for systems with a large number of DOF.
- Analytical calculation. While this method is very efficient during run-time, cumbersome analytical derivation of all applied forces has to be performed in a preprocessing step.

Vehicle dynamics simulations contain terms with high nonlinearities, hence analytical calculation is chosen to compute the Jacobian matrix. Equation (2.123) contains all elements that have to be computed analytically. While $\frac{\partial}{\partial \boldsymbol{x}} (\boldsymbol{Z}\boldsymbol{z})$ can be computed easily, see fig. 2.16, Jacobian calculation of the applied forces $\boldsymbol{f}_a$ is more complex. In [2] it is shown that the calculation for applied forces can be simplified by introduction of intermediate variables. For instance the suspension spring force with stiffness c and length l_0 is realized as a 1D-force element with intermediate variable $\boldsymbol{d}$ as

$$\boldsymbol{F} = c\left(|\boldsymbol{d}| - l_0\right) \frac{\boldsymbol{d}}{|\boldsymbol{d}|} \ . \tag{5.1}$$

The gradient then follows as

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}} \ . \tag{5.2}$$

Hereby $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}$ can be computed analytically and $\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}$ yields the same equations as the sensitivity analysis of the connection rod constraint, see section 2.4.3. This method is used extensively for the Jacobian matrix calculation of the applied forces.
Some simplifications can be made due to the choice of reference frame for flexible bodies and due to the Coriolis and centrifugal force behavior. In [54] it is shown that for the Buckens reference frame (see [57]) the terms $\boldsymbol{c}_i, \boldsymbol{C}_i^t$ vanish and $\boldsymbol{C}_i^r = \boldsymbol{C}_{i1}^r \boldsymbol{q}_i$. This simplifies the calculation of mass matrix, Coriolis and centrifugal forces and the Jacobian calculation for these properties. The Jacobian entries for Coriolis and centrifugal forces can be neglected entirely due to the smooth behavior in vehicle dynamics simulations, see [2].
The Jacobian matrices for all applied forces are used for the baseline index-2 simulation model and all flexible entries for the mass matrix and Coriolis and centrifugal forces are calculated. An additional model with these simplifications employed is used within the performance evaluation in section 5.4.

5.1.3 Index-1 Vehicle Model

The index-2 *Simulink* model can be altered to state the EOM on index-1 level. This results in a more expensive calculation of the constraint equations but does

not require a Jacobian calculation. For time integration of the EOM on index-1 level, eq. (2.188) is used with eq. (2.94) to calculate the values of $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda} = \left(\boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{G}^T\right)^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\boldsymbol{\omega} - \boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{f}_a - \beta^2\boldsymbol{g} - 2\alpha\frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}t}\right) . \tag{5.3}$$

This allows the explicit formulation of the state derivatives as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{Z}\boldsymbol{z}\\ \dot{\boldsymbol{z}} &= \boldsymbol{M}^{-1}\left(\boldsymbol{f}_a - \boldsymbol{G}^T\boldsymbol{\lambda}\right) .\end{aligned} \tag{5.4}$$

The assembly of the EOM is shown in figure A.8, which is then used in a modified *Simulink* model, see figure A.7, to be integrated using built-in integrators. Due to the elastic coordinates with low damping and high eigenfrequencies the explicit 4th order RK method is used for FMBS preferably, see section 2.2.2. Besides the RT capable index-1 time integration this model allows the usage of step-size and error controlled integrators for non-RT simulations using the built-in *Simulink* solvers.

5.1.4 Generalized Subsystems to Decrease the Calculation Effort

In section 2.1.9 the ODE formulation of MBS in tree structures is derived. However, the vehicle model described in this section contains kinematic loops involving a flexible body and hence it is generally not possible to resolve the whole system of EOM in ODE form. Out of the 41 constraints necessary for the vehicle model, 20 are required to constrain the wheels to the wheel carriers. A wheel is connected only to its corresponding wheel carrier, with a rotational joint and hence only one relative DOF. The wheels are not involved in any kinematic loops, so the topology of the vehicle model can be changed such that the wheel rotational motion is described by a generalized rotational DOF, see figure 5.4.

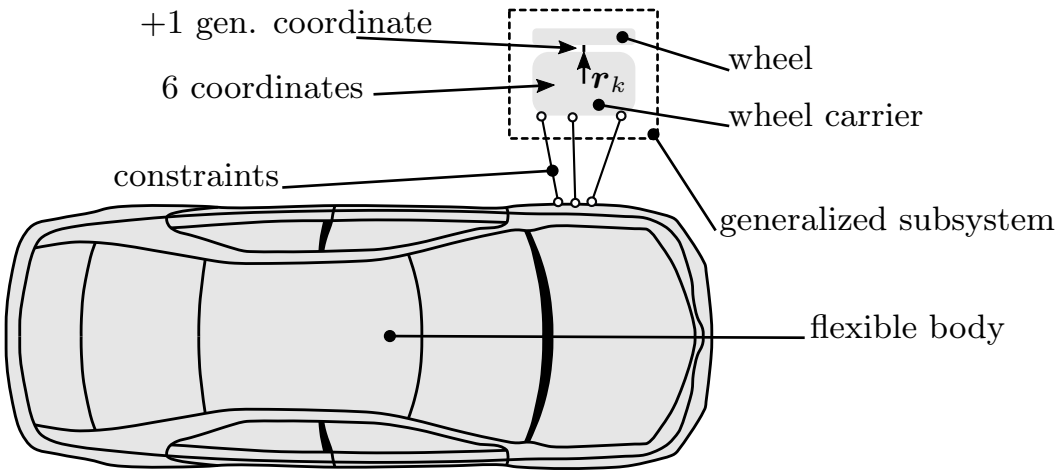


Figure 5.4: Topology of the model with generalized subsystems.

For a generalized subsystem i describing wheel carrier and wheel the set of position and velocity coordinates is chosen as

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{\rho}_{\mathrm{WC}}^T & \boldsymbol{p}_{\mathrm{WC}}^T & \alpha_{\mathrm{Wh}} \end{bmatrix}^T \tag{5.5}$$

$$\boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{v}_{\mathrm{WC}}^T & {}^i\boldsymbol{\omega}_{\mathrm{WC}}^T & \dot{\alpha}_{\mathrm{Wh}} \end{bmatrix}^T \ , \tag{5.6}$$

which are cartesian coordinates for the wheel carrier extended by $\alpha_{\mathrm{Wh}}, \dot{\alpha}_{\mathrm{Wh}}$. The vector $\boldsymbol{r}_k$ denotes the position vector from the wheel carrier reference system to the wheel rotation axis point located at the wheel CoG. The wheel rotation axis is assumed to be aligned with the y-axis of the wheel carrier reference system. This gives the local Jacobian matrices for Jourdain's principle of eq. (2.100)

$$\begin{aligned} \boldsymbol{J}_{\mathrm{WC}}^t &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{J}_{\mathrm{WC}}^r &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{E} & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{J}_{\mathrm{Wh}}^t &= \begin{bmatrix} \boldsymbol{E} & \widetilde{\boldsymbol{r}}_k & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{J}_{\mathrm{Wh}}^r &= \begin{bmatrix} \boldsymbol{E} & \boldsymbol{A}_{\alpha_{\mathrm{Wh}}} & [0\ 1\ 0]^T \end{bmatrix} \end{aligned}$$

and the global Jacobian matrix for a wheel carrier - wheel combination

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{\mathrm{WC}}^t \\ \boldsymbol{J}_{\mathrm{WC}}^r \\ \boldsymbol{J}_{\mathrm{Wh}}^t \\ \boldsymbol{J}_{\mathrm{Wh}}^r \end{bmatrix} \ . \tag{5.7}$$

Insertion in eq. (2.100) results in a generalized subsystem of only 8 position coordinates. Since the tree of this subsystem consists of only two bodies and one relative coordinate, the calculation effort increase is moderate.
The generalized EOM for each generalized system then can be calculated and combined with the other bodies of the system to a vehicle model reduced by 24 coordinates and 24 constraint equations.

5.2 Validation of the MATLAB/Simulink Model With MBS Software

To ensure the correctness of the RT vehicle dynamics simulations developed in *MATLAB/Simulink*, a common driving maneuver is compared to the results of the commercially available MBS software *ADAMS*. To eliminate the influence of flexible body formulations, only rigid body models are used for this test. Therefore, all elastic DOF are removed from the *Simulink* model, all parameters of the model are equal in both models. *ADAMS* provides a MF-tire model interface, which was used and parametrized equal to the *Simulink* index-2 model. The driving maneuver chosen for this test is constant cornering along a relatively small circle of ≈22.5 m radius, induced by a steering rod translation step of 15 mm after 3 s straight movement along the global x-axis. The maneuver is driven with an initial speed of 40 $\frac{\mathrm{km}}{\mathrm{h}}$ and kept constant by a PI-controller applying a driving

torque at the front wheels. To compare the simulations, global position and suspension travel is compared and shown in figure 5.5, figure 5.6 and figure 5.7.

The global position of both simulations are almost coincident, which validates identical behavior of both models since no position or trajectory control is present. The suspension travel is a good measure for lateral behavior of the vehicle body and correct kinematics of the suspension and shows the same static deflection in both simulations. Since the initial condition calculation is different in *ADAMS*, the *ADAMS* simulation requires around 2 s to settle to the same deflection as the *Simulink* simulation, see figure 5.6 and figure 5.7.
The *ADAMS* simulation shows a slightly different deflection at the beginning of cornering, which may be caused by the tire model implemented in *ADAMS* and the stabilized index-1 formulation in combination with the *GSTIFF* solver of *ADAMS*. Although the software uses the MF-tire model shown in section 4.3.2, it has to be treated as a blackbox model since the exact implementation is not known. However, since the static suspension deflection is identical, the *Simulink* model is presumed to behave equivalent to the *ADAMS* model.

5.3 Results of Driving Maneuvers

The model validated in the previous section is equipped with a flexible chassis and several driving maneuvers are performed to demonstrate the influence of structural flexibility. The first test is a washboard excitation, similar to the tests performed in [6]. The second test is a double lane change similar to the ISO 3888-2 test [28].

5.3.1 Washboard Track

The washboard track is used in vehicle development to ensure the dynamic stability of vehicle components under low frequency load. The track is a straight road section with six segments of equally distributed bumps of decreasing distance. The washboard track is passed with a constant speed of 50 $\frac{\text{km}}{\text{h}}$ for this test. The vehicle speed is kept constant during the test by a PI-controller applying the torque at the front wheels. The road bumps lead to a step-like excitation of the wheels in a frequency range of 11–20 Hz at 50 $\frac{\text{km}}{\text{h}}$.
To get a general idea how the loads on the flexible car body look like, figure 5.8 shows the acceleration signal of the rear left (RL) suspension dome. The essential shape of the acceleration result is comparable to the results shown in [6]. However, since the loads are largely dependent on the parametrization of the vehicle model, specifically the suspension damper setup, the acceleration magnitude may differ from a real vehicle measurement.
The acceleration peaks show little difference between the rigid and flexible body simulation since the acceleration is largely caused by rigid body motion. To further investigate the difference between rigid and elastic simulation, figure 5.9

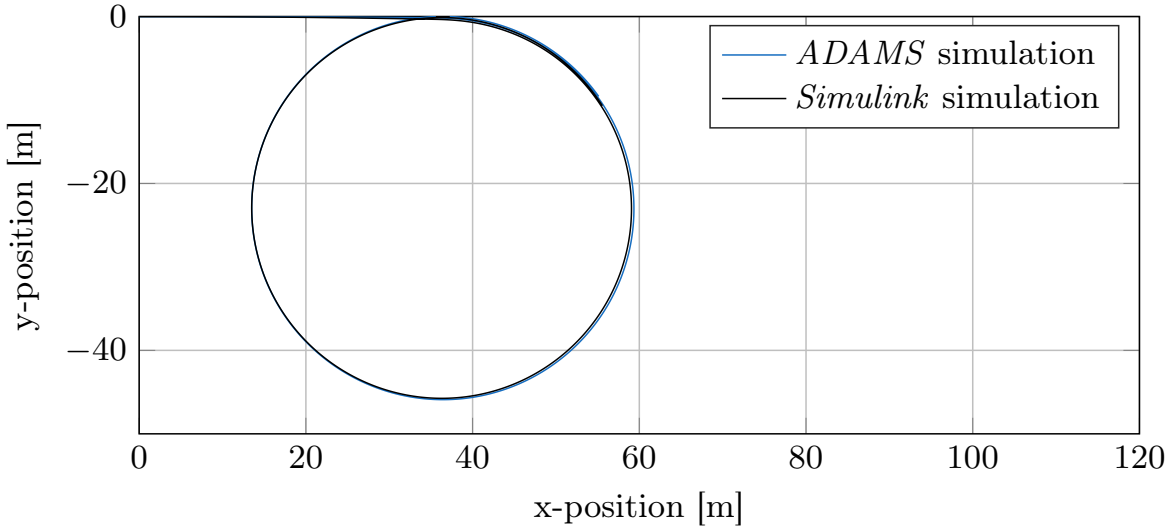


Figure 5.5: Global position comparison, car body CoG.

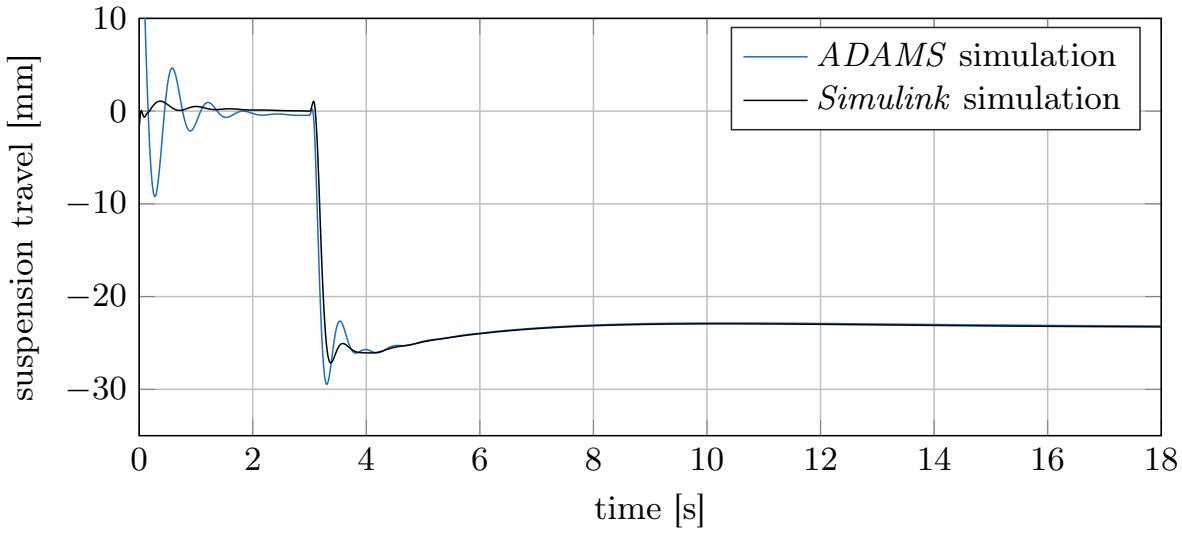


Figure 5.6: Spring deflection, front left.

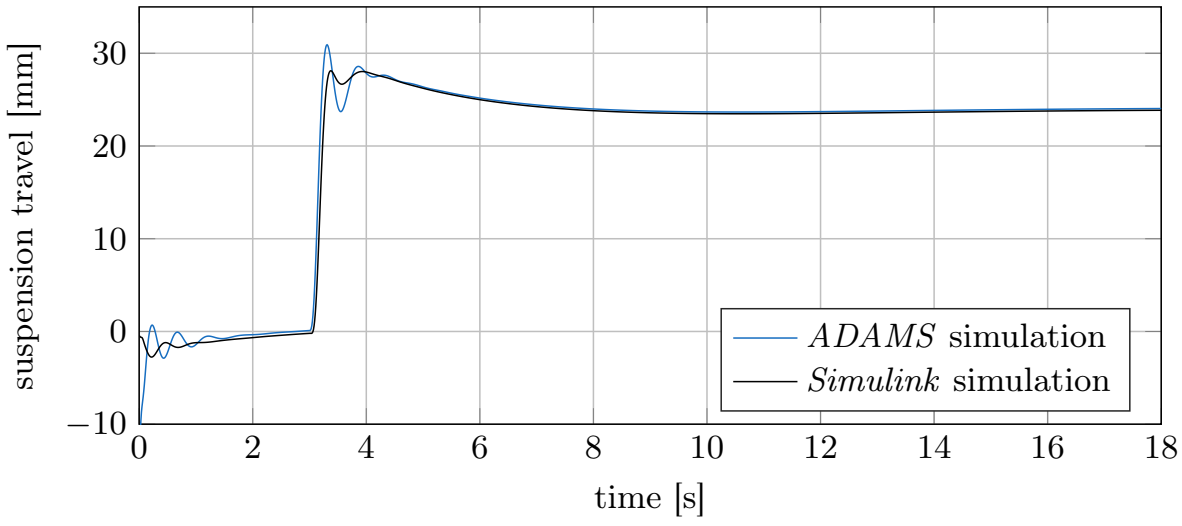


Figure 5.7: Spring deflection, rear right.

and figure 5.10 show the power spectral densities (PSD) of the car body CoG and suspension dome acceleration signal. The car body CoG PSD is almost identical since the FFoR is located at the car body CoG. For the suspension dome there is almost no difference for the excitation frequencies of the washboard track, but for >20 Hz the flexible simulation shows significantly higher intensity than the rigid simulation.
Figure 5.12 shows the PSD of the remaining three suspension domes. The front suspension domes show less difference between rigid and flexible simulation, which indicates that the front structure of the car is stiffer than the rear structure.
The dimensionless elastic coordinates displayed in figure 5.11 show that the flexible body is indeed excited by the road bumps, although the magnitude is dependent on the excitation frequency. Furthermore, the static deflection caused by gravitation can be seen as well. It is worth mentioning that the shape functions are orthonormalized, hence no constrained mode or eigenmode is directly related to the coordinates shown.
On page 98 a time lapse view of the first washboard bumps is shown. As a result from the model order reduction the elastic deformation of all FE nodes can be calculated and visualized. The color range is set from blue (0 mm) to red (1 mm) deformation. The line tripods on top of the vehicle represent the tire forces at the specific time step.
The vehicle obviously shows a static deformation, yielding the front and rear end of the vehicle to be always deformed with respect to the vehicle reference frame. The bumps result in a significantly larger deformation that is slightly delayed compared to the excitation, which can be seen at the frames between 0.86 s and 0.90 s.

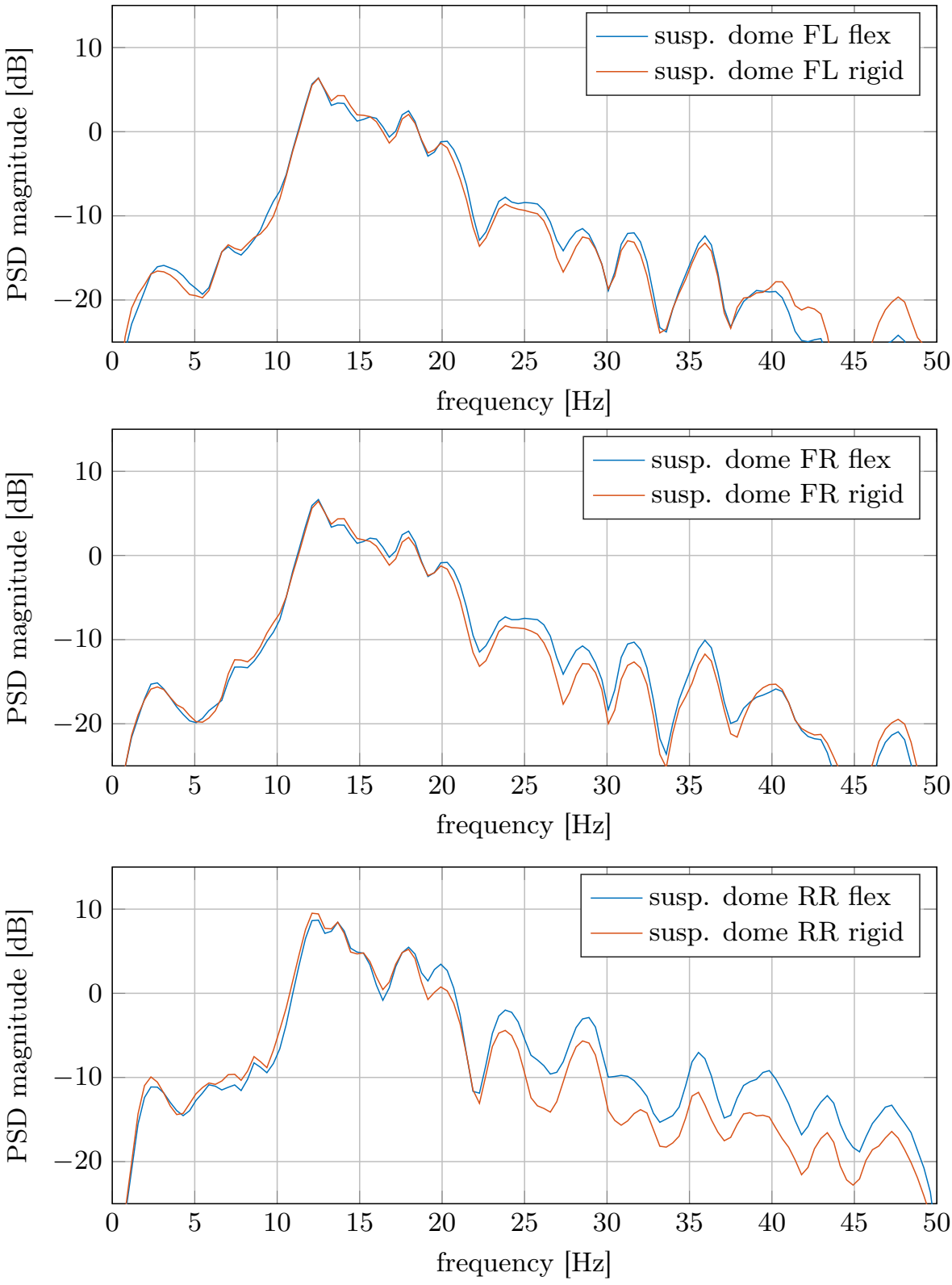
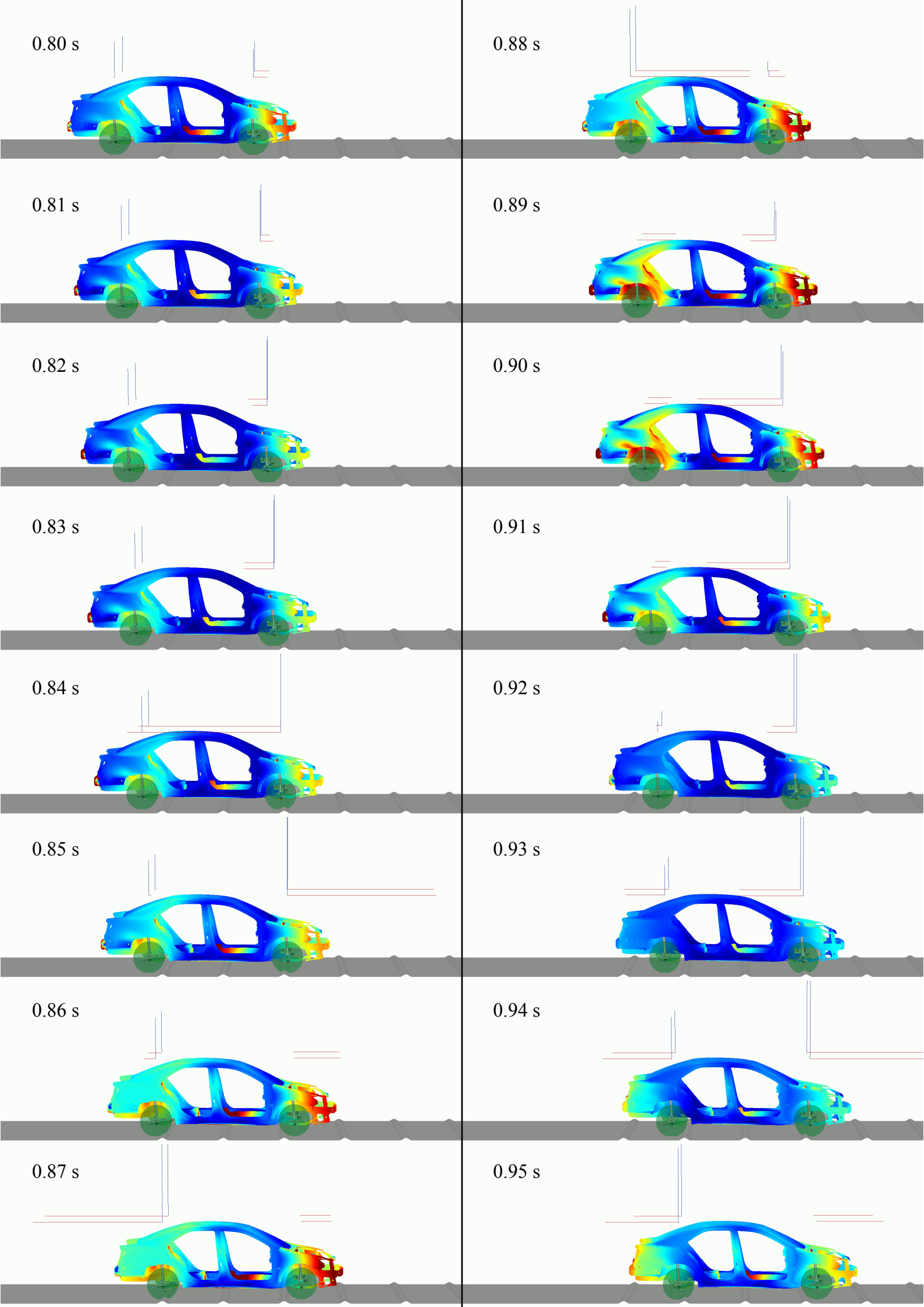


Figure 5.12: Power spectral densities of suspension dome FL, FR and RR.


0.80 s
0.88 s
0.81 s
0.89 s
0.82 s
0.90 s
0.83 s
0.91 s
0.84 s
0.92 s
0.85 s
0.93 s
0.86 s
0.94 s
0.87 s
0.95 s

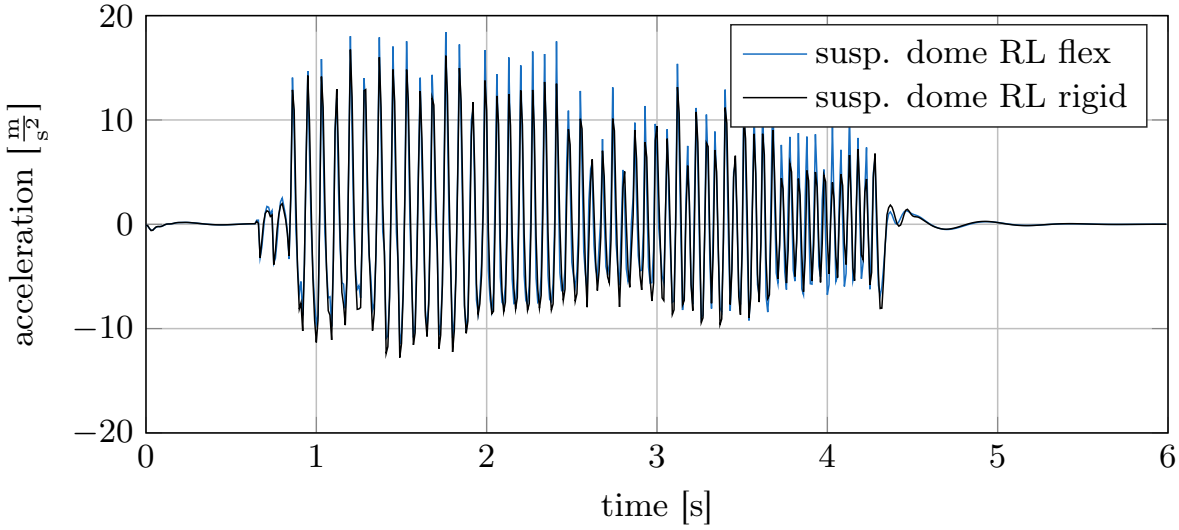


Figure 5.8: Acceleration of suspension dome RL.

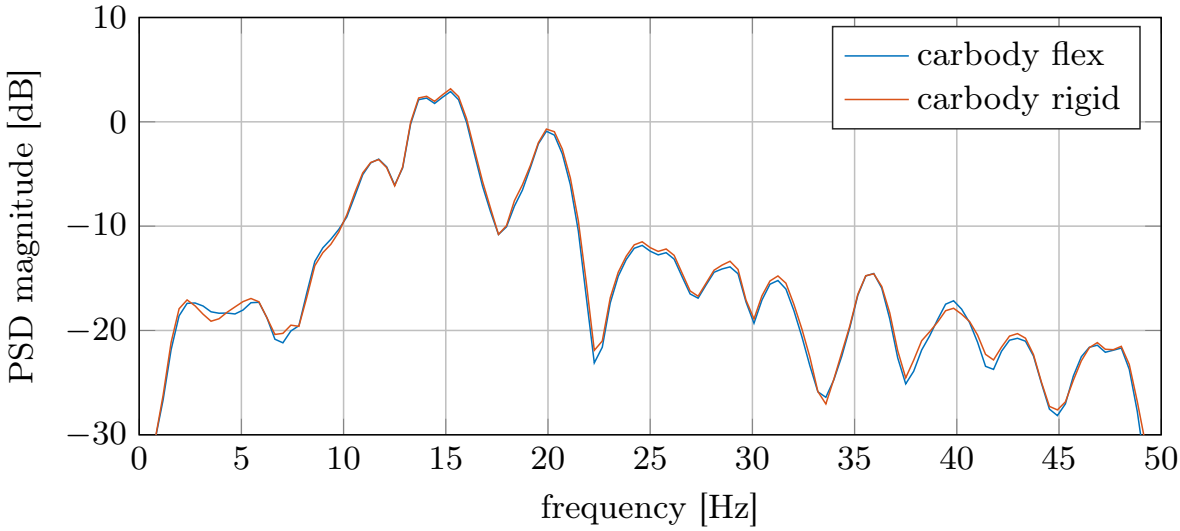


Figure 5.9: Power spectral density of car body CoG.

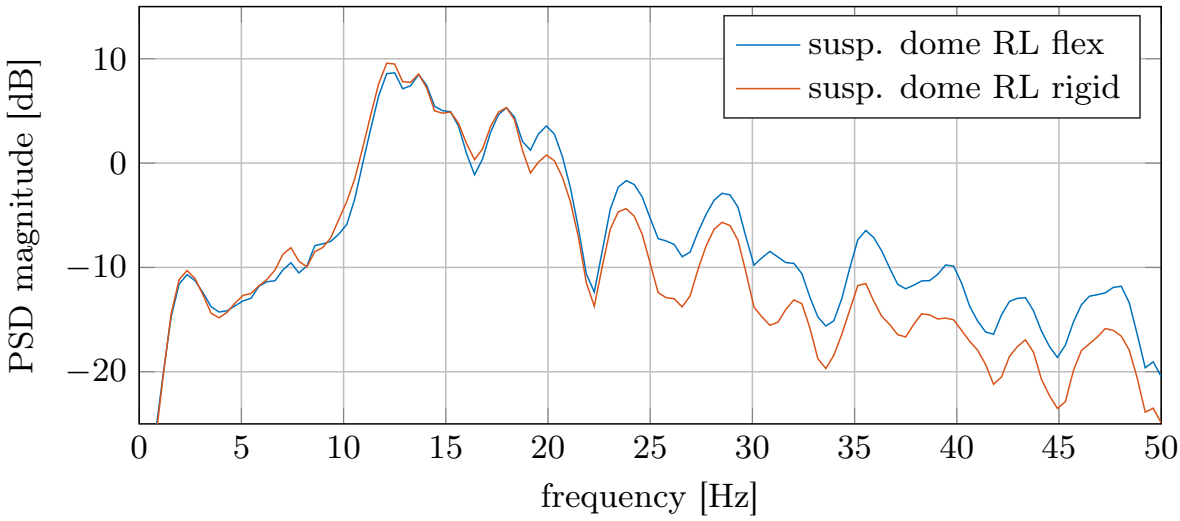
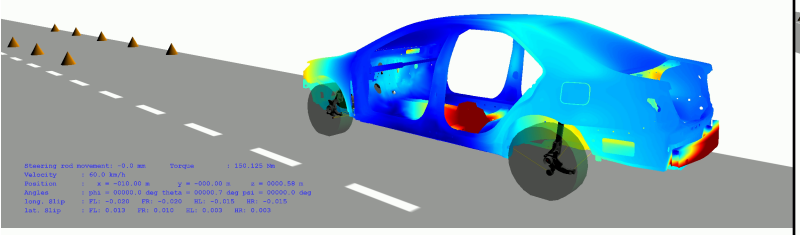


Figure 5.10: Power spectral density of suspension dome RL.

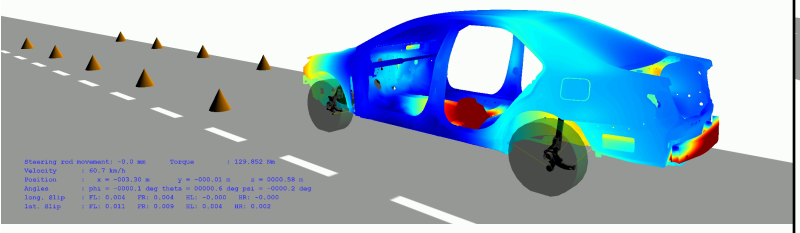
5.3.2 Double Lane Change

The double lane change - also known as goose test or obstacle avoidance test - is a high speed maneuver, in which the vehicle has to change the lane within a given distance and go back after the obstacle is passed. No engine torque is applied during this maneuver. The lane change is performed with both rigid and flexible test under identical conditions and identical steering control input. The steering input is adjusted so that the rigid model stays within the lane for an initial speed of 60 $\frac{\mathrm{km}}{\mathrm{h}}$.
The trajectory of both the rigid and the flexible model is shown in figure 5.13. Both models behave almost identically, although the influence of the flexible body can be seen at the end of the maneuver, when the flexible model trajectory deviated by around 0.3 m in lateral direction. This can be interpreted with the elastic body that introduces an additional torsional stiffness of the car body. The torsional stiffness acts as an additional compliance of the front right suspension while taking the abrupt left turn at the beginning of the maneuver.
Figure 5.14 shows the elastic coordinates with the highest absolute amplitude during the lane change. The influence can be seen as well in the visualization of the maneuver on page 101. Additionally the vehicle positions at the time frames are marked in figure 5.13. The car body displays the intuitive deformation dependent on the load shift from right to left side and back.

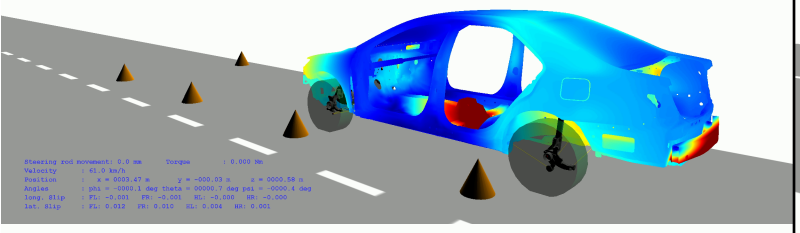
0.00 s



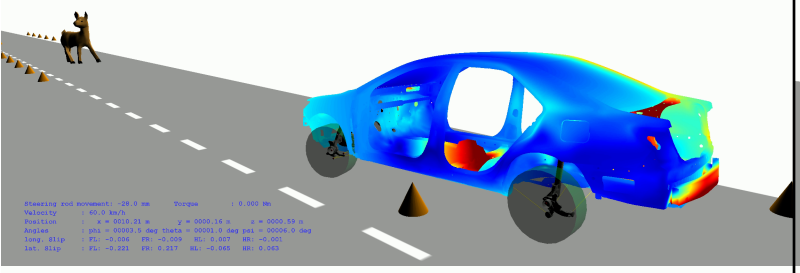
0.40 s



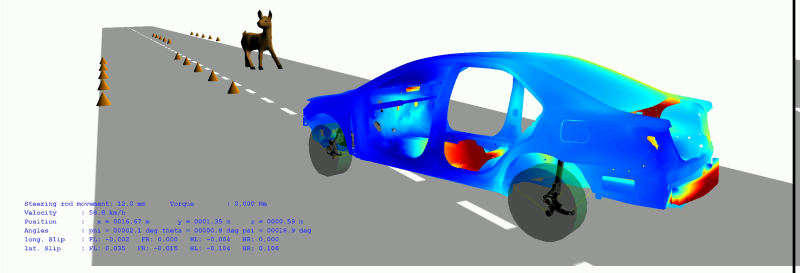
0.80 s



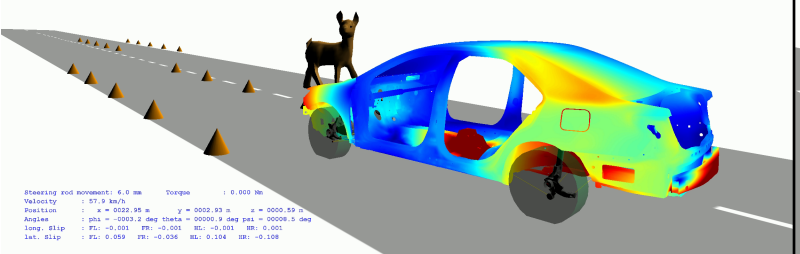
1.20 s



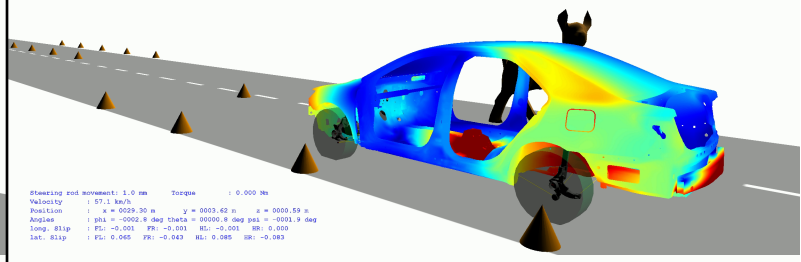
1.60 s



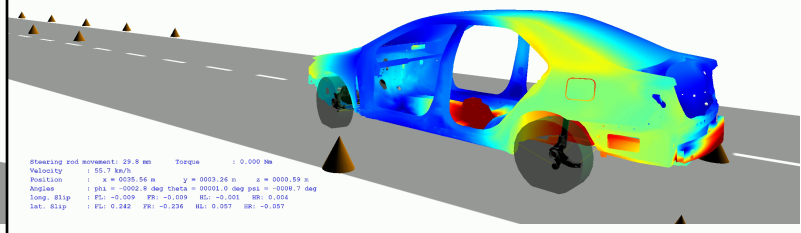
2.00 s



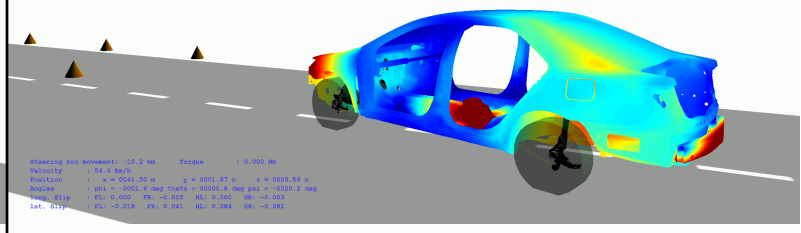
2.40 s



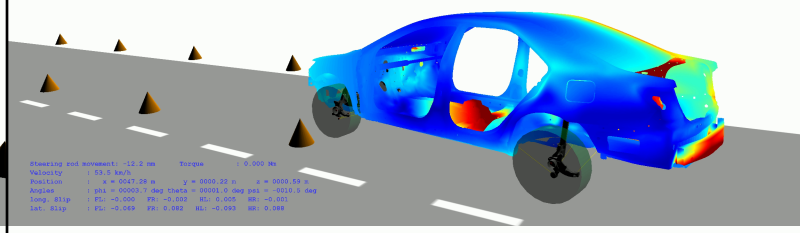
2.80 s



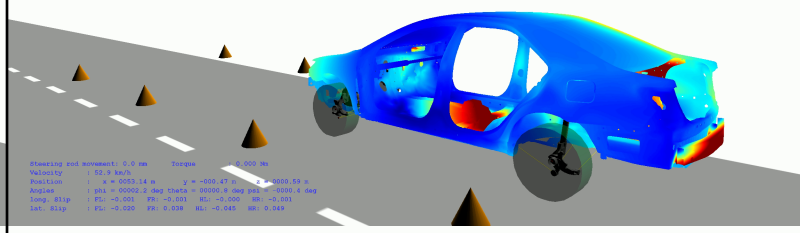
3.20 s



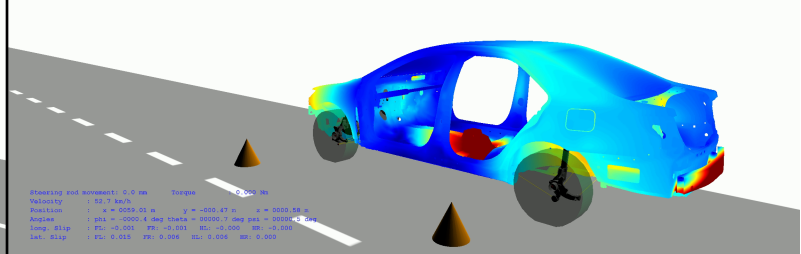
3.60 s



4.00 s



4.40 s



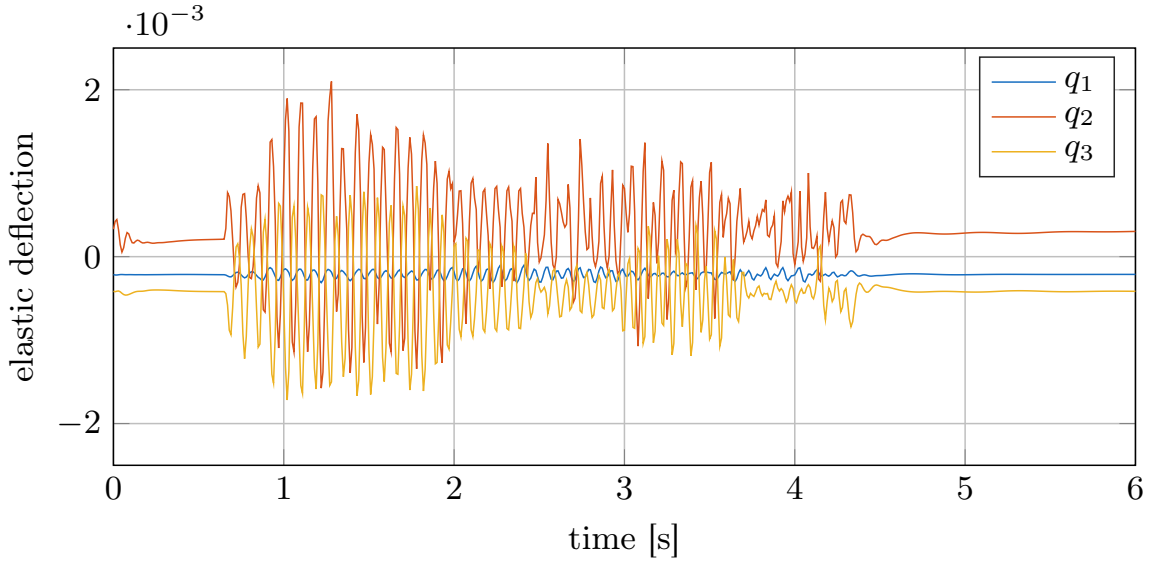


Figure 5.11: Elastic coordinates.

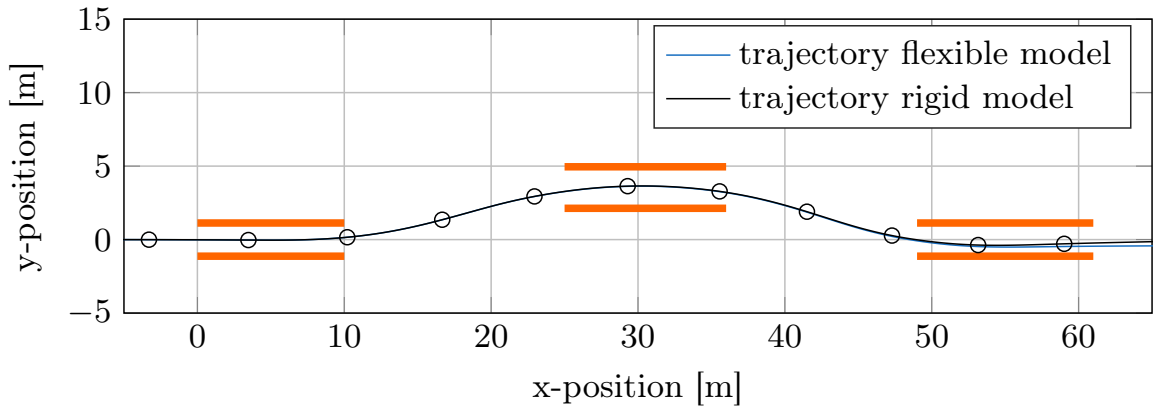


Figure 5.13: Trajectories of double lane change.

5.4 Performance Evaluation

A large-scale simulation model is required to use the existing calculation power of the simulation platform efficiently in order to be executed in real-time. In section 2.2 several conditions for RT capability are defined and all *Simulink* simulation models shown in this chapter are RT capable in theory since their calculation load is calculable in advance. However, the most crucial point is the amount of calculation time necessary to perform the simulation. To quantify the RT capability, the real-time factor of the simulation models is determined. It is defined as

$$\text{RT factor} = \frac{\text{execution time}}{\text{simulation time}} , \tag{5.8}$$

which means that a RT factor < 1 marks a RT capable model since the CPU time necessary to simulate the model is less than the wall-clock time.
In addition to the models shown in section 5.1 the index-2 model with a rigid vehicle body is included in the comparison. For the performance comparison the

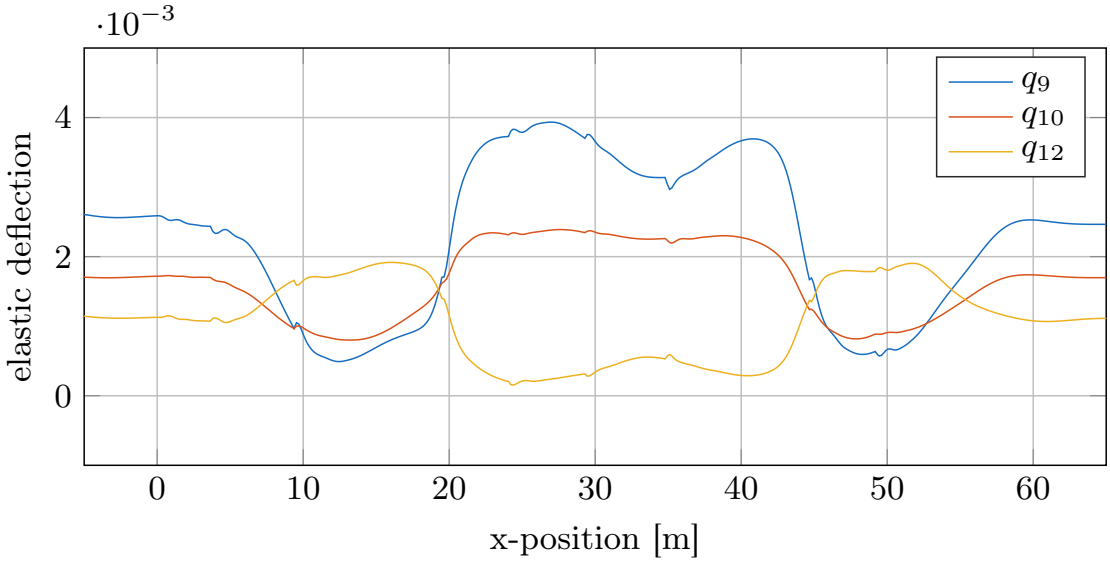


Figure 5.14: Selected elastic coordinates.

vehicle is going along a straight line with a speed of 60 $\frac{\text{km}}{\text{h}}$, traveling approximately 1700 m. The driving maneuver performed during the performance run does not affect the performance results since all calculations of the vehicle model are independent of the current vehicle state. Figure 5.15 shows the real-time factors for several performance runs on a quad-core 3.6 GHz CPU, averaged over 25 runs of 100 s simulation time. Figure 5.16 compares the trajectories of all models and shows that all models behave almost identically in the simulations.

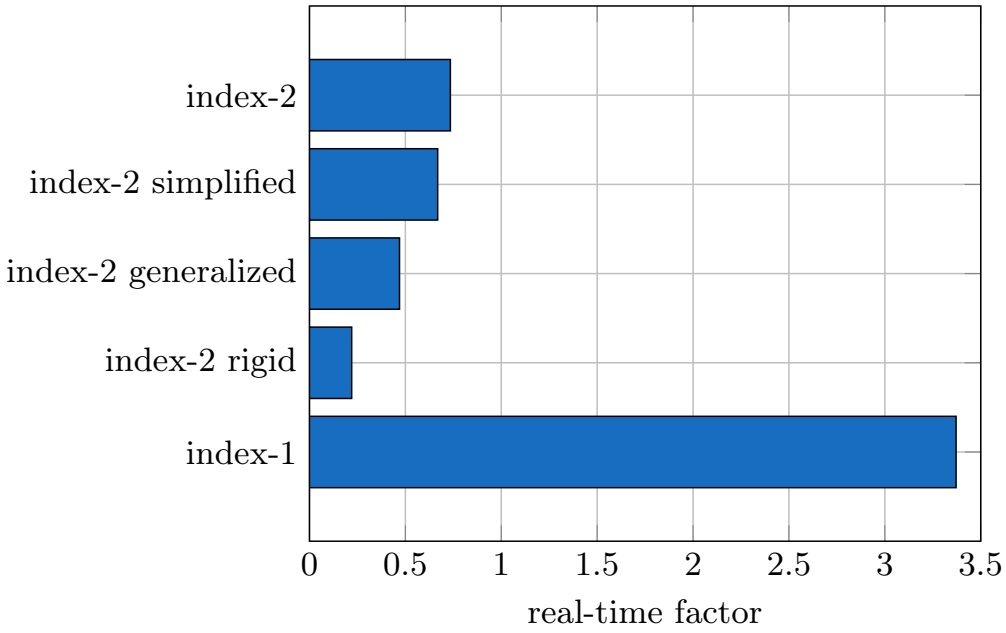


Figure 5.15: RT performance of *Simulink* models.

All index-2 simulations are performed with the linearly-implicit Euler method, a step size of 10 ms and constraint stabilization with the projection method. The

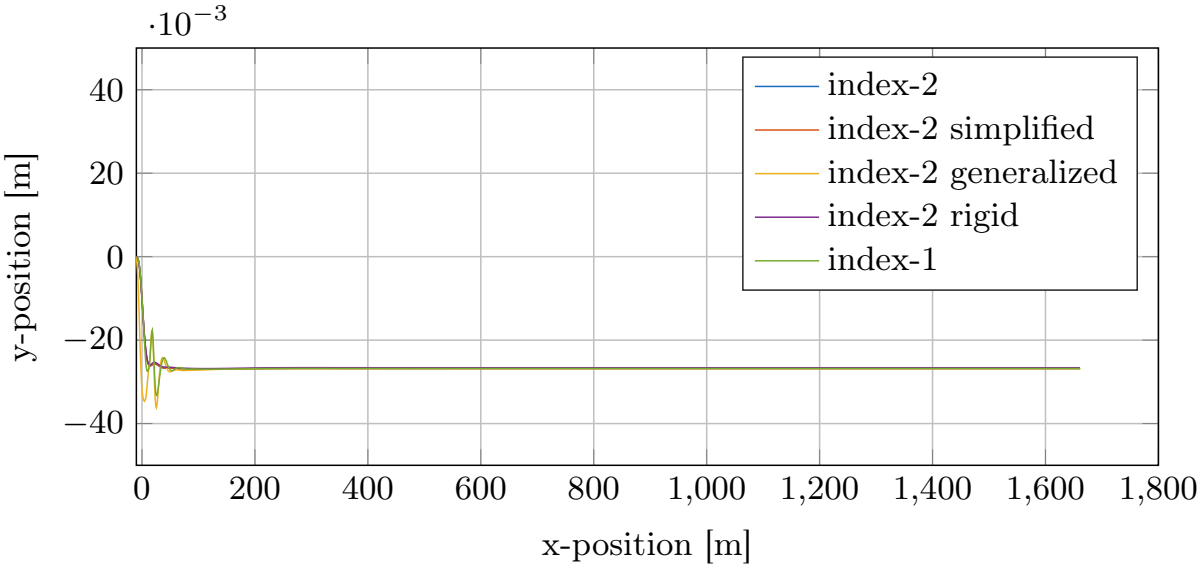


Figure 5.16: Vehicle CoG trajectories of performance simulations.

index-1 simulation is executed by the built-in 4th order Runge-Kutta solver of *Simulink*, a step size of 1 ms and Baumgarte stabilization. Larger step sizes or lower order solvers yield to numerical instability of the index-1 simulation, hence this can be seen as the performance limit for explicit time integration.
The advantage of the index-2 formulation is obvious compared to the index-1 formulation of the identical simulation setup. While the index-1 formulation is not RT capable with a real-time factor > 3, the index-2 formulation requires only 22 % of the simulation time with a RT factor of 0.73. Simplification of Jacobians and removing unnecessary calculations of the flexible body formulation reduces the simulation time by 9 %, and combining a generalized formulation of the suspension setup decreases the simulation time by 36 %.
Compared to the flexible index-2 model a rigid body simulation takes only 30 % of the simulation time. Although the number of DOF is reduced by only 40 %, the simulation time decreases by 70 %, which is caused by the significantly more complex evaluation of inertial properties, Jacobian calculation and constraint sensitivity analysis for flexible bodies.
Overall index-2 formulation of flexible MBS is proven to be a suitable method for real-time simulations. Compared to index-1 formulation, the increase of computational effort for Jacobian calculation for the index-2 models is more than compensated by the option to choose significantly larger step sizes and less effort for the constraint evaluation. However, considering structural flexibility in RT simulations still is very time consuming and since the number of elastic coordinates is virtually unlimited, the currently available computation power can be exceeded easily.

Conclusions and Outlook

The present work reveals interesting aspects on the formulation of structural flexibilities in RT applications. Based on efficient and RT capable flexible MBS definitions a vehicle model based on a modern road vehicle is successfully simulated in real-time.

Various aspects of the basics on flexible MBS are covered first. After a brief summary of the basics on rigid MBS kinematics, quaternions as a rotation formulation for rigid and flexible bodies are introduced and the advantages of the straight-forward sensitivity analysis are shown. Selected principles of mechanics are covered as well, necessary to derive the EOM for flexible MBS in cartesian and generalized coordinates.

Real-time capable solution of MBS in general requires numerical solution of DAE if kinematic loops are present. Index reduction can be used to transform a system of DAE into a system of stiff ODE, stating the algebraic constraint equations on acceleration level. This allows the application of explicit integration methods. The group of RT capable Runge-Kutta methods is presented along with a numerical stability analysis of the solvers. To solve the DAE with constraint equations stated on velocity level, the linearly-implicit Euler method is introduced. Numerical errors for the position constraints may occur when solving the EOM in an index-reduced form. While constraint stabilization usually is performed with iterative correction methods, an iteration-free, RT capable constraint stabilization process is introduced to prevent numerical drift.

Due to the integration methods used for RT simulation, applying the standard floating frame of reference approach yields to numerical inefficiencies. A mixed reference frame approach is introduced, defining translational DOF in the global reference system and rotational and elastic DOF in the body-related reference frame. The structural data of elastic bodies is stored in the standard input data (SID) format. This data format is retained for the mixed reference frame approach and the EOM for flexible MBS are formulated so that the standard SID structure for bodies defined for the FFoR approach still can be used.

Constraint definition for flexible MBS is performed using the mixed reference frame approach and defining all constraint equations in global coordinates, which is again advantageous for RT capable integration methods. The constraint equations are derived on velocity and acceleration level to assemble systems of DAE on index-2 and index-1 level.

Besides using quaternions for rotation parametrization, quaternion derivatives can be employed to state the EOM of flexible MBS completely without the use of angular velocity. The EOM for quaternion derivatives are derived and applied to a minimal example, proving the functionality of the concept. However, since this formulation results in higher system dimensions and hence slightly longer simulation times, the approach is not employed to a full vehicle model.
The methods on efficient RT simulation of flexible MBS is applied to a full vehicle model with a car body modeled as a flexible body. A detailed vehicle model for crash simulation purposes of a 2012 Toyota Camry is reduced to the body in white, containing more than 1 million elements. Component mode synthesis (CMS) methods are used to reduce the vehicle structure to an elastic body containing only 40 elastic DOF. Interface nodes at the suspension attachment points and engine bearings are retained for the flexible multibody simulation. By applying the well-known Magic Formula (MF) tire model and OpenCRG road model a RT capable vehicle model with 10 bodies is assembled.
Modeling and simulation of the vehicle model is realized with the numerical simulation environment *MATLAB/Simulink*. Several driving maneuvers are performed to demonstrate the validity and performance of the formulation of RT capable flexible MBS. The commercial software *ADAMS* is used to verify the simulation model in *Simulink*, which yields to identical results for the dynamic behavior of the suspension kinematics in steady-state cornering. A washboard track creates a harsh, low-frequency excitation of the vehicle body when driving it with constant velocity. The vehicle model with structural flexibility is compared with a rigid model to emphasize the influence of the elastic body, which can be pointed out clearly with the frequency response of the interface node acceleration. Double lane-change maneuvers are used to determine the limits of dynamic lateral vehicle stability. The deformation behavior during such a maneuver is shown as well, yielding a slightly different trajectory of the flexible model due to the deformation of the vehicle body.
In a performance comparison the index-2 formulation shows a clear advantage towards an index-1 formulation of flexible MBS. Despite the need for Jacobian calculation, the larger integration step size and the less costly constraint formulation yields an approximate 75 % decrease in simulation time. Jacobian calculation can be simplified due to the smooth behavior of Coriolis and centrifugal forces, and the Buckens reference frame allows a simplified calculation of elastic properties. This results in a 9 % decrease in simulation time. Combining generalized subsystems with DAE formulation results in a 36 % decrease in simulation time. On a modern desktop CPU a real-time factor of less than 0.5 is achieved.

6.1 Outlook

The methods presented in this work are proven as very efficient for the RT capable simulation of flexible MBS. The application example of a full vehicle simulation demonstrates the usability of RT capable flexible MBS. However, the

methods are not limited to vehicle models but may also be applied to other fields of use. Aircraft modeling, marine applications or robotics are possible scenarios for flexible multibody models of similar structure. Real-time capable models can be employed to control purposes as well e.g. to actively dampen or eliminate structural vibrations.
Real-time capability is verified by ensuring the real-time factor to be kept < 1 on a desktop computer system. Desktop operating systems are not RT systems because the task scheduling and execution priority is determined by the operating system. As a subsequent step the model deployment to RT systems like Hardware-in-the-Loop (HiL) platforms or Driver-in-the-Loop (DiL) systems would further verify the RT capability and quantify the performance limits under realistic RT conditions.
The simulation model shown in the vehicle dynamics simulations is based on FE data and exemplary tire and road data. However, to integrate such a simulation model into a vehicle development process, many model parameters like suspension characteristics, damping constants or bushing behavior have to be matched to real vehicle data. In [55] a parameter adjustment procedure is presented that uses optimization methods to fit the model behavior to a set of test data.
While application to various fields of mechanical engineering is obvious, the methods shown in this work may also be used for Virtual Reality (VR) or entertainment purposes. Quaternions as a rotation parametrization is nowadays a quasi-standard in visualization [24], which recommends the parallel use in physics and visualization software. The low real-time factor of the simplified vehicle models even allows the simulation on devices with low computational power like mobile devices or system on a chip (SOC) environments.

BIBLIOGRAPHY

[1] M. Arnold, B. Burgermeister, and A. Eichberger. Linearly implicit time integration methods in real-time applications: Daes and stiff odes. *Multibody System Dynamics*, 17(2-3):99–117, 2007.

[2] M. Arnold, B. Burgermeister, C. Führer, G. Hippmann, and G. Rill. Numerical methods in vehicle system dynamics: state of the art and current developments. *Vehicle System Dynamics*, 49(7):1159–1207, 2011.

[3] K.-J. Bathe. *Finite Element Procedures*. K.-J. Bathe, Watertown, MA, 2006.

[4] O. A. Bauchau. *Flexible Multibody Dynamics*. Springer Science+Business Media B.V, 2011.

[5] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16, 1972.

[6] R. Benz. *Fahrzeugsimulation zur Zuverlässigkeitsabsicherung von karosseriefesten Kfz-Komponenten*. PhD thesis, Universität Karlsruhe (TH), 2008.

[7] D. Bestle. *Analyse und Optimierung von Mehrkörpersystemen: Grundlagen und rechnergestützte Methoden*. Springer-Verlag, 1994.

[8] P. Betsch and R. Siebert. Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration. *International Journal for Numerical Methods in Engineering*, 79(4):444–473, 2009.

[9] B. Burgermeister, M. Arnold, and B. Esterl. DAE time integration for real-time applications in multi-body dynamics. *ZAMM - Journal of Applied Mathematics and Mechanics*, 86(10):759–771, 2006.

[10] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.

[11] Center for Collision Safety and Analysis. CCSA webpage, retrieved 22.11.2018. `http://www.ccsa.gmu.edu/`.

[12] R. Craig and M. Bampton. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1313–1319, 1968.

[13] J. Cuadrado, J. Cardenal, and E. Bayo. Modeling and solution methods for efficient real-time simulation of multibody dynamics. *Multibody System Dynamics*, 1(3):259–280, 1997.

[14] J. Cuadrado, D. Dopico, M. A. Naya, and M. Gonzalez. Real-time multibody dynamics and applications. In *Simulation Techniques for Applied Dynamics*, pages 247–311. Springer, 2008.

[15] J. Cuadrado, U. Lugris, D. Dopico, and M. Gonzalez. Topological vs global penalty formulations for real-time flexible multibody dynamics. *Journal of System Design and Dynamics*, 1(3):422–433, 2007.

[16] J. G. de Jalon and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems: The Real Time Challenge*. Springer-Verlag, 1994.

[17] L. Euler. De integratione aequationum differentialium per approximationem. *Opera Omnia*, 11:424, 1913.

[18] J. Fehr. *Automated and Error Controlled Model Reduction in Elastic Multibody Systems*. PhD thesis, University of Stuttgart, 2011.

[19] J. Fehr, M. Burkhardt, and P. Eberhard. Oblique model reduction techniques for the simulation of the nvh-relevant states of a racing kart. *Proceedings of the 7th European Nonlinear Dynamics Conference 2011*, 2011.

[20] G. Fichera, M. Lacagnina, and F. Petrone. Modelling of torsion beam rear suspension by using multibody method. *Multibody System Dynamics*, 12(4):303–316, 2004.

[21] M. Gipser. Reifenmodelle für Komfort- und Schlechtwegesimulationen. In *7. Aachener Kolloquium für Fahrzeug- und Motorentechnik*, pages 519–540, 1998.

[22] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, 2002.

[23] W. R. Hamilton. *Elements of Quaternions*, volume 1. Longmans, Green, and Company, 1899.

[24] A. J. Hanson. *Visualizing Quaternions*. Elsevier, 2006.

[25] E. J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems*, volume 1. Allyn and Bacon, 1989.

[26] A. Held. *On Structural Optimization of Flexible Multibody Systems*. PhD thesis, University of Stuttgart, 2014.

[27] E. G. Hemingway and O. M. O'Reilly. Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics*, 44(1):31–56, 2018.

[28] ISO 3888-2:2011. Passenger cars – Test track for a severe lane-change manoeuvre – Part 2: Obstacle avoidance. Standard, International Organization for Standardization, Geneva, Switzerland, May 2011.

[29] ISO/IEC 2382:2015. Information technology - Vocabulary. Standard, International Organization for Standardization, Geneva, Switzerland, May 2015.

[30] B. Kessler. *Bewegungsgleichungen für Echtzeitanwendungen in der Fahrzeugdynamik*. PhD thesis, Universität Stuttgart, 1989.

[31] H. Kopetz. *Real-time Systems: Design Principles for Distributed Embedded Applications*. Springer Science & Business Media, 2011.

[32] M. Lehner. *Modellreduktion in elastischen Mehrkörpersystemen*. PhD thesis, Universität Stuttgart, 2007.

[33] D. Marzougui, C.-D. Kan, and N. E. Bedewi. Development and validation of an NCAP simulation using LS-DYNA 3D. Technical report, FHWA/NHTSA National Crash Analysis Center, The George Washington University, 1996.

[34] G. Mastinu and M. Ploechl. *Road and Off-Road Vehicle System Dynamics Handbook*. CRC Press, 2014.

[35] L. Morello, L. R. Rossini, G. Pia, and A. Tonoli. *The Automotive Body Volume II: System Design*. Springer Science & Business Media, Dordrecht, 2011.

[36] MSC Software. *Adams 2014 Online Help*, 2014.

[37] F. Naets, T. Tamarozzi, G. H. K. Heirman, and W. Desmet. Real-time flexible multibody simulation with global modal parameterization. *Multibody System Dynamics*, 27(3):267–284, 2012.

[38] P. E. Nikravesh. *Computer-Aided Analysis of Mechanical Systems*. Prentice-Hall, Inc., 1988.

[39] P. E. Nikravesh, R. A. Wehage, and O. K. Kwon. Euler parameters in computational kinematics and dynamics. part 1. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(3):358–365, 1985.

[40] H. Pacejka and I. Besselink. *Tire and Vehicle Dynamics*. Elsevier, third edition, 2012.

[41] K. Popp and W. Schiehlen. *Ground Vehicle Dynamics*. Springer Science & Business Media, 2010.

[42] R. Reichert, P. Mohan, D. Marzougui, C. D. Kan, and D. Brown. Validation of a toyota camry finite element model for multiple impact configurations. *SAE International*, 2016.

[43] P. Riekert and T.-E. Schunck. Zur Fahrmechanik des gummibereiften Kraftfahrzeugs. *Ingenieur-Archiv*, 11(3):210–224, 1940.

[44] G. Rill. *Road Vehicle Dynamics: Fundamentals and Modeling*. CRC Press, 2011.

[45] G. Rill and T. Schaeffer. *Grundlagen und Methodik der Mehrkörpersimulation*. Springer Vieweg, second edition, 2014.

[46] R. E. Roberson and R. Schwertassek. *Dynamics of Multibody Systems*. Springer-Verlag, 1988.

[47] W. Rulka and E. Pankiewicz. MBS approach to generate equations of motions for hil-simulations in vehicle dynamics. *Multibody System Dynamics*, 14(3):367–386, 2005.

[48] E. Sampo, A. Sorniotti, and A. Crocombe. Chassis torsional stiffness: Analysis of the influence on vehicle dynamics. *SAE International*, 2010.

[49] W. Schiehlen and P. Eberhard. *Applied Dynamics*. Springer International Publishing, 2014.

[50] A. Schmid and S. Förschl. Vom realen zum virtuellen Reifen - Reifenmodellparametrierung. *ATZ*, 111(03), 2009.

[51] A. Schmitt, H. Grossert, and R. Seifried. Evaluation and experimental validation of efficient simulation models for optimization of an electrical formula car. In *Volume 6: 14th International Conference on Multibody Systems, Nonlinear Dynamics, and Control*. American Society of Mechanical Engineers, ASME, aug 2018.

[52] A. Schmitt and R. Seifried. Real-time simulation of elastic multibody systems with application in vehicle dynamics. *PAMM*, 15(1):75–76, 2015.

[53] A. Schmitt and R. Seifried. Comparison of various models and integration method for real-time simulation of complex vehicle models with structural flexibility. In *Proceedings of ISMA2016 International Conference on Noise and Vibration Engineering*, 2016.

[54] A. Schmitt and R. Seifried. Performance aspects of real-time capable flexible multibody simulations. In *Proceedings of the 8th ECCOMAS Thematic Conference on Multibody Dynamics*, 2017.

[55] A. Schmitt, R. Seifried, K. Sandmann, and B. Zillmann. Efficient vehicle simulation to calculate acceleration loads on electric vehicle batteries. In *45. Tagung des DVM-Arbeitskreises Betriebsfestigkeit: Effiziente Auslegung und Absicherung in der Betriebsfestigkeit*, pages 19–30, 2018.

[56] D. Schramm, M. Hiller, and R. Bardini. *Vehicle Dynamics*. Springer-Verlag, 2014.

[57] R. Schwertassek and O. Wallrapp. *Dynamik flexibler Mehrkörpersysteme*. Vieweg, 1999.

[58] R. Seifried. *Dynamics of Underactuated Multibody Systems*. Springer International Publishing, 2014.

[59] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, third edition, 2005.

[60] A. A. Shabana. *Computational Continuum Mechanics*. Cambridge University Press, first edition, 2008.

[61] M. D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

[62] J. Sobieszczanski-Sobieski, S. Kodiyalam, and R. Y. Yang. Optimization of car body under constraints of noise, vibration, and harshness (nvh), and crash. *Structural and Multidisciplinary Optimization*, 22(4):295–306, 2001.

[63] A. Soppa. *Krylov-Unterraum-basierte Modellreduktion zur Simulation von Werkzeugmaschinen*. PhD thesis, Technical University of Braunschweig, 2011.

[64] H. Sugiyama, A. A. Shabana, M. A. Omar, and W.-Y. Loh. Development of nonlinear elastic leaf spring model for multibody vehicle systems. *Computer Methods in Applied Mechanics and Engineering*, 195(50-51):6925–6941, 2006.

[65] The Mathworks Inc. *MATLAB Documentation, Release R2017a*, 2017.

[66] VIRES Simulationstechnologie GmbH. OpenCRG webpage, retrieved 17.11.2018. `http://www.opencrg.org`.

[67] O. Wallrapp. Standardization of flexible body modeling in multibody system codes, part 1: Definition of standard input data. *Journal of Structural Mechanics*, 22(3):283–304, 1994.

[68] H. Yamashita, P. Jayakumar, and H. Sugiyama. Physics-based flexible tire model integrated with lugre tire friction for transient braking and cornering analysis. *Journal of Computational and Nonlinear Dynamics*, 11(3):031017, 2016.

Acronyms

CMS	component mode synthesis.
CoG	center of gravity.
DAE	differential-algebraic equation.
DOF	degree of freedom.
EOM	equation of motion.
FE	finite element.
FFoR	floating frame of reference.
FMBS	flexible multibody system.
HiL	Hardware-in-the-Loop.
MBS	multibody system.
NVH	noise, vibration and harshness.
ODE	ordinary differential equation.
RK	Runge-Kutta.
RT	real-time.
SID	standard input data.

Additional Information on Simulation Models

This chapter contains additional pictures, numerical data and results of the 2012 Toyota Camry simulation model.

A.1 Constraint Setup

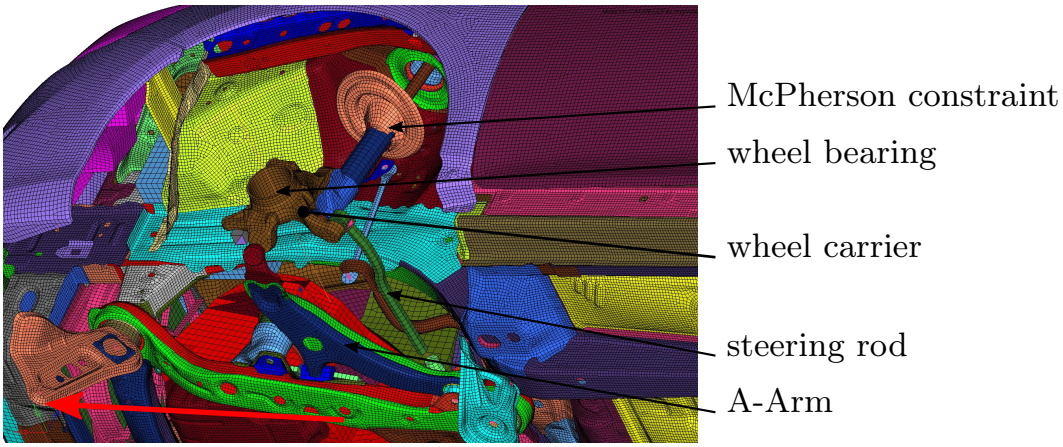


Figure A.1: Front suspension in FE model (Tire removed)

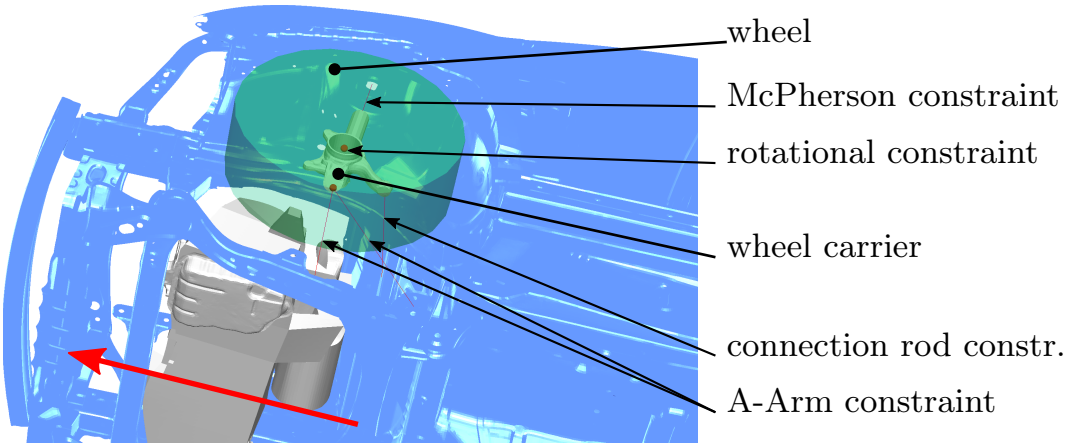


Figure A.2: Front suspension modeling in VR environment

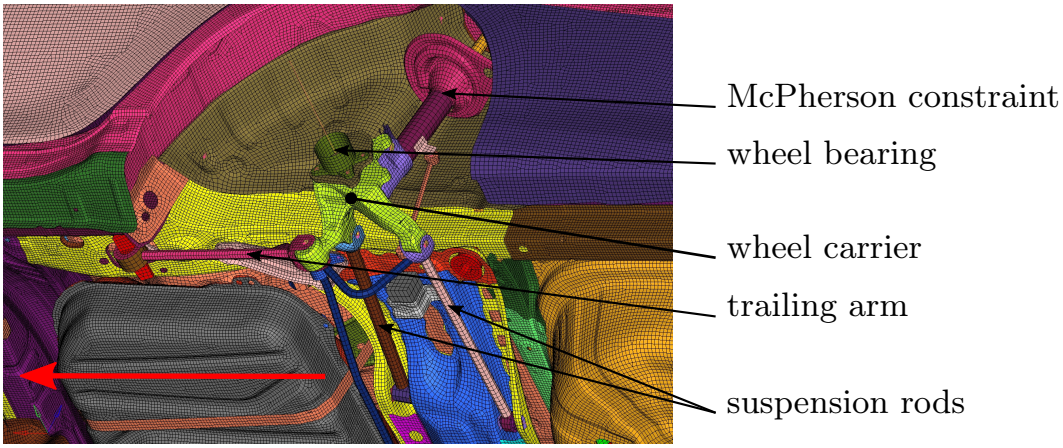


Figure A.3: Rear suspension in FE model (Tire removed)

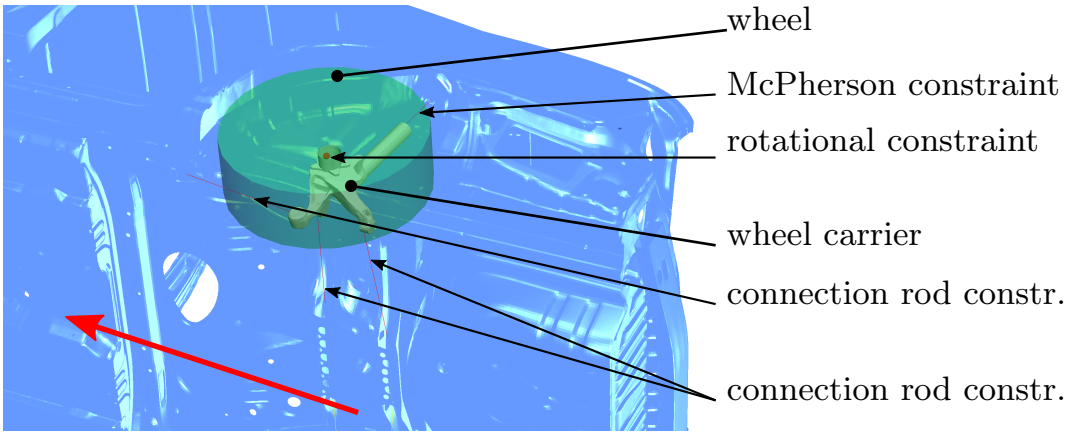


Figure A.4: Rear suspension modeling in VR environment

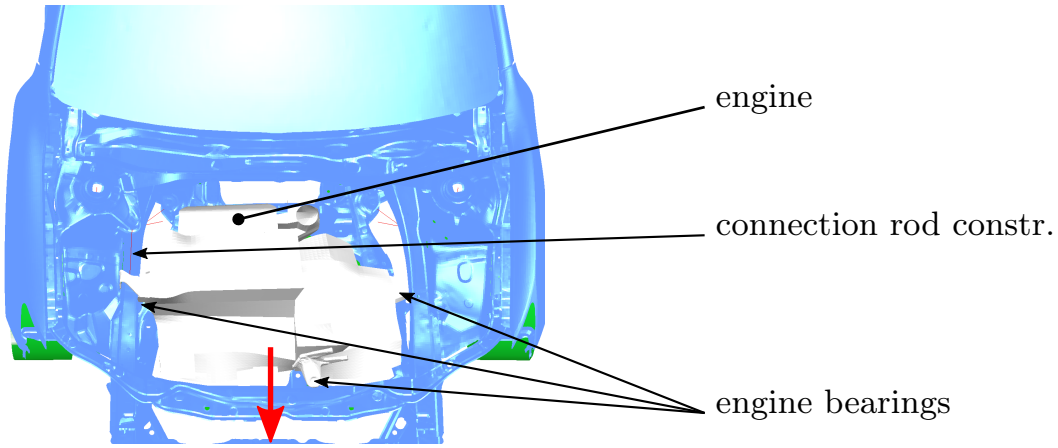


Figure A.5: Engine modeling in VR environment

A.2 Numerical Data

Table A.1: Force element properties.

Force element	Type	C	D	Coordinates of nodes
suspension FL	1D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$1.93\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} 1.369 & 0.599 & 0.330 \end{bmatrix}$ m WC FL: $\begin{bmatrix} -0.023 & -0.147 & 0.314 \end{bmatrix}$ m
suspension FR	1D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$1.93\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} 1.369 & -0.599 & 0.330 \end{bmatrix}$ m WC FR: $\begin{bmatrix} -0.023 & 0.147 & 0.314 \end{bmatrix}$ m
suspension RL	1D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$1.93\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} -1.432 & 0.592 & 0.318 \end{bmatrix}$ m WC RL: $\begin{bmatrix} -0.032 & -0.139 & 0.312 \end{bmatrix}$ m
suspension RR	1D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$1.93\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} -1.432 & -0.592 & 0.317 \end{bmatrix}$ m WC RR: $\begin{bmatrix} -0.032 & 0.139 & 0.312 \end{bmatrix}$ m
engine mount L	3D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$5\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} 1.507 & 0.490 & -0.214 \end{bmatrix}$ m engine: $\begin{bmatrix} -0.051 & 0.490 & -0.152 \end{bmatrix}$ m
engine mount C	3D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$5\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} 1.876 & 0.130 & -0.128 \end{bmatrix}$ m engine: $\begin{bmatrix} 0.318 & 0.130 & -0.066 \end{bmatrix}$ m
engine mount R	3D	$50\ \frac{\mathrm{kN}}{\mathrm{m}}$	$5\ \frac{\mathrm{kNs}}{\mathrm{m}}$	car body: $\begin{bmatrix} 1.510 & -0.480 & -0.208 \end{bmatrix}$ m engine: $\begin{bmatrix} -0.048 & -0.480 & -0.146 \end{bmatrix}$ m

Table A.2: Constraint types and location (relative to body CoG).

Constraint	Type	Coordinates of nodes	DOF
A-arm FL	connection rod	car body: $[\ 1.419 \quad 0.419 \quad -0.355]$ m WC FL: $[\ 0.009 \quad -0.009 \quad -0.116]$ m	1
A-arm FL	connection rod	car body: $[\ 1.133 \quad 0.368 \quad -0.335]$ m WC FL: $[\ 0.009 \quad -0.009 \quad -0.116]$ m	1
steering rod FL	connection rod	car body: $[\ 1.234 \quad 0.380 \quad -0.255]$ m WC FL: $[-0.129 \quad -0.026 \quad -0.038]$ m	1
McPherson strut FL	McPherson constr.	car body: $[\ 1.369 \quad 0.599 \quad 0.330]$ m WC FL: $[-0.023 \quad -0.147 \quad 0.314]$ m	2
wheel bearing FL	rotational constr.	WC FL: $[\ 0.000 \quad 0.400 \quad 0.000]$ m wheel FL: $[\ 0.000 \quad 0.000 \quad 0.000]$ m	5
A-arm FR	connection rod	car body: $[\ 1.419 \quad -0.417 \quad -0.356]$ m WC FR: $[\ 0.009 \quad 0.009 \quad -0.116]$ m	1
A-arm FR	connection rod	car body: $[\ 1.135 \quad -0.365 \quad -0.335]$ m WC FR: $[\ 0.009 \quad 0.009 \quad -0.116]$ m	1

Table A.2: Constraint types and location (relative to body CoG).(contd.)

Constraint	Type	Coordinates of nodes	DOF
steering rod FR	connection rod	car body: $[\ 1.229 \quad -0.385 \quad -0.259]$ m WC FR: $[-0.129 \quad 0.026 \quad -0.038]$ m	1
McPherson strut FR	McPherson constr.	car body: $[\ 1.369 \quad -0.599 \quad 0.330]$ m WC FR: $[-0.023 \quad 0.147 \quad 0.314]$ m	2
wheel bearing FR	rotational constr.	CB FR: $[\ 0.000 \quad -0.400 \quad 0.000]$ m wheel FR: $[\ 0.000 \quad 0.000 \quad 0.000]$ m	5
suspension strut RL 1	connection rod	car body: $[-0.892 \quad 0.600 \quad -0.305]$ m WC RL: $[\ 0.010 \quad -0.110 \quad -0.072]$ m	1
suspension strut RL 2	connection rod	car body: $[-1.317 \quad 0.199 \quad -0.283]$ m WC RL: $[-0.150 \quad -0.113 \quad -0.099]$ m	1
tension strut RL	connection rod	car body: $[-1.536 \quad 0.144 \quad -0.307]$ m WC RL: $[\ 0.040 \quad -0.085 \quad -0.139]$ m	1
McPherson strut RL	McPherson constr.	car body: $[-1.432 \quad 0.592 \quad 0.318]$ m WC RL: $[-0.032 \quad -0.139 \quad 0.312]$ m	2

Table A.2: Constraint types and location (relative to body CoG).(contd.)

Constraint	Type	Coordinates of nodes	DOF
wheel bearing RL	rotational constr.	WC RL: $\begin{bmatrix} 0.000 & 0.400 & 0.000 \end{bmatrix}$ m wheel RL: $\begin{bmatrix} 0.000 & 0.000 & 0.000 \end{bmatrix}$ m	5
suspension strut RR 1	connection rod	car body: $\begin{bmatrix} -0.888 & -0.601 & -0.306 \end{bmatrix}$ m WC RR: $\begin{bmatrix} 0.010 & 0.110 & -0.072 \end{bmatrix}$ m	1
suspension strut RR 2	connection rod	car body: $\begin{bmatrix} -1.318 & -0.200 & -0.284 \end{bmatrix}$ m WC RR: $\begin{bmatrix} -0.150 & 0.113 & -0.099 \end{bmatrix}$ m	1
tension strut RR	connection rod	car body: $\begin{bmatrix} -1.537 & -0.146 & -0.307 \end{bmatrix}$ m WC RR: $\begin{bmatrix} 0.040 & 0.085 & -0.139 \end{bmatrix}$ m	1
McPherson strut RR	McPherson constr.	car body: $\begin{bmatrix} -1.432 & -0.592 & 0.317 \end{bmatrix}$ m WC RR: $\begin{bmatrix} -0.032 & 0.139 & 0.312 \end{bmatrix}$ m	2
wheel bearing RR	rotational constr.	WC RR: $\begin{bmatrix} 0.000 & -0.400 & 0.000 \end{bmatrix}$ m wheel RR: $\begin{bmatrix} 0.000 & 0.000 & 0.000 \end{bmatrix}$ m	5
engine torque support	connection rod	car body: $\begin{bmatrix} 1.470 & -0.482 & 0.265 \end{bmatrix}$ m engine: $\begin{bmatrix} 0.066 & -0.485 & 0.280 \end{bmatrix}$ m	1

A.3 Additional Simulink Views

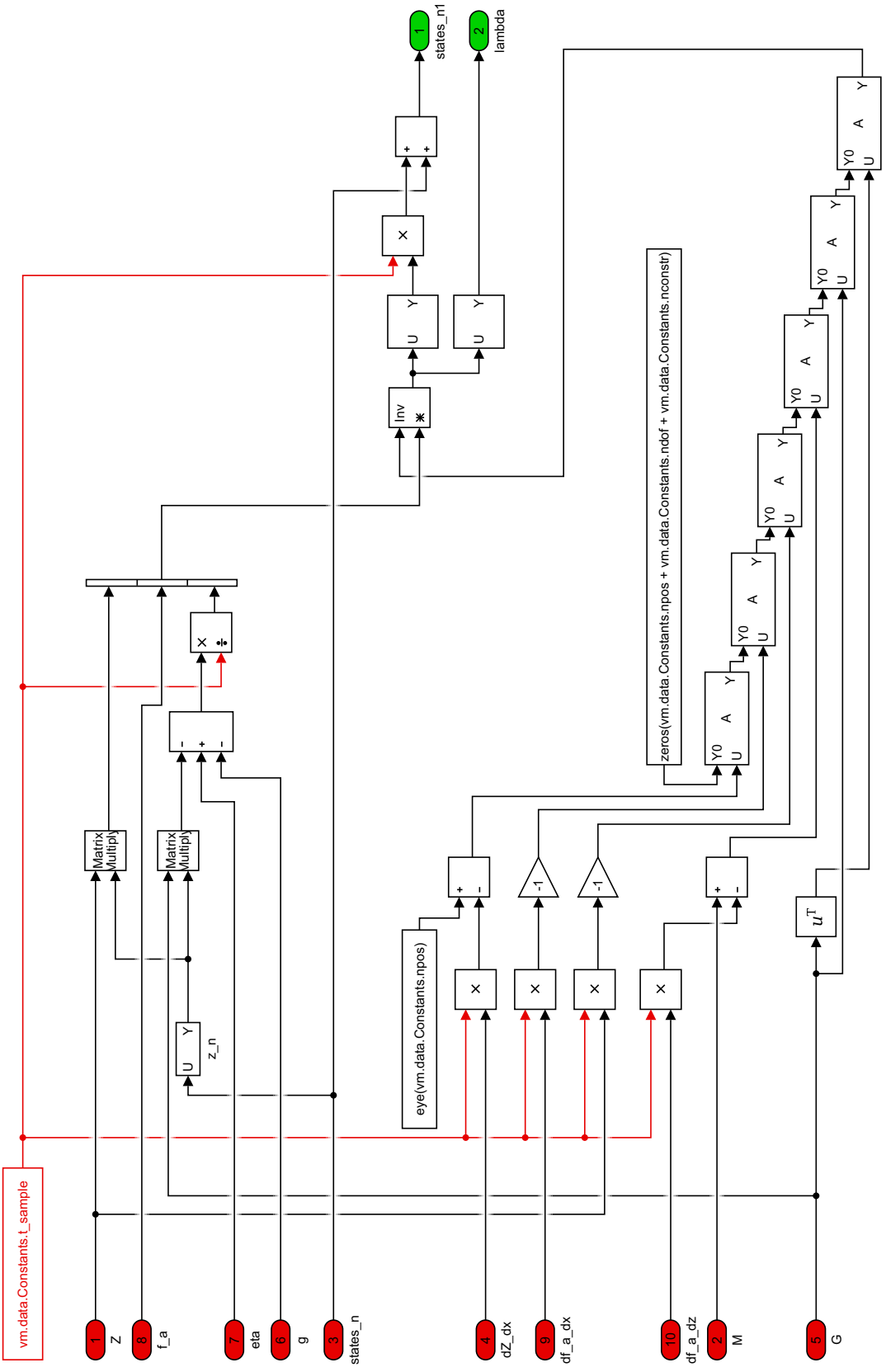


Figure A.6: Index-2 integration scheme.

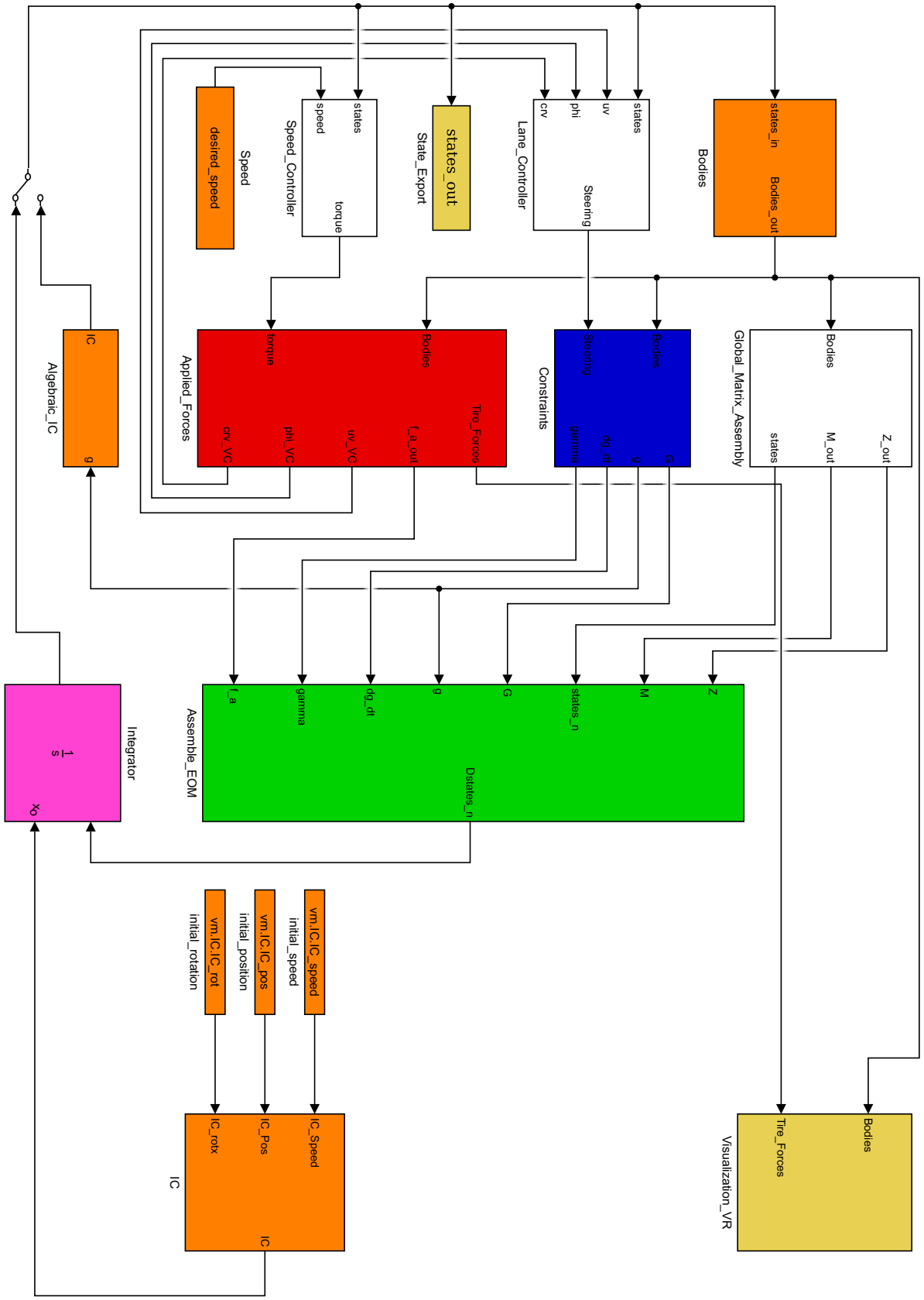


Figure A.7: Top level layout of the index-1 *Simulink* model.

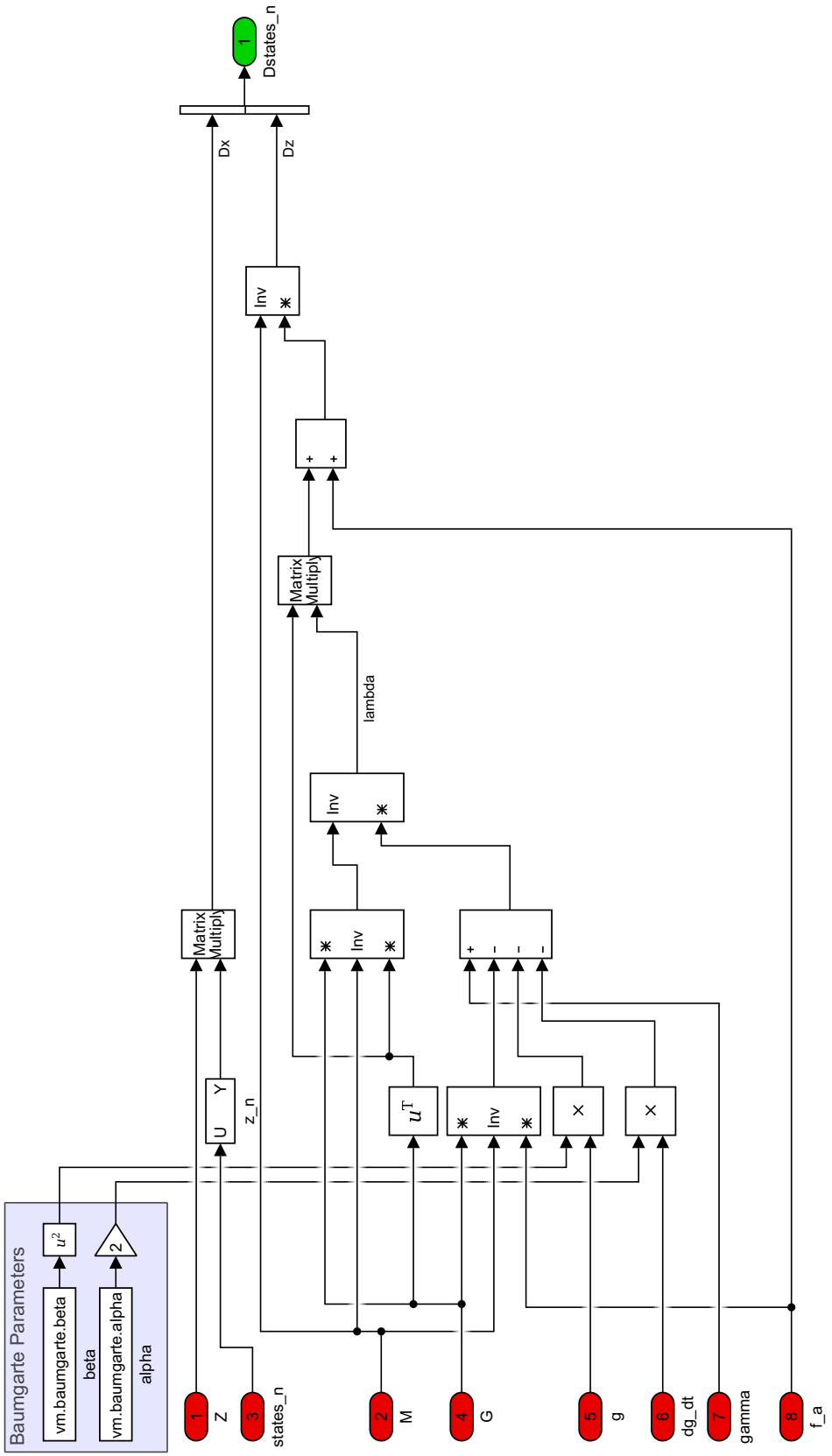


Figure A.8: Index-1 EOM assembly.