Computational Methods in Applied Sciences

Zdravko Terze *Editor*

Multibody Dynamics

Computational Methods and Applications

ECCOMAS
European Community on Computational Methods in Applied Sciences

Springer

Computational Methods in Applied Sciences

Volume 35

Series editor

Eugenio Oñate, Barcelona, Spain

For further volumes:
http://www.springer.com/series/6899

Zdravko Terze
Editor

Multibody Dynamics

Computational Methods and Applications

Springer

Editor
Zdravko Terze
Faculty of Mechanical Engineering
and Naval Architecture
University of Zagreb
Zagreb
Croatia

ISSN 1871-3033
ISBN 978-3-319-07259-3 ISBN 978-3-319-07260-9 (eBook)
DOI 10.1007/978-3-319-07260-9
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014942454

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

To meet the challenges of the fast development of new technologies, many areas of contemporary engineering and applied sciences, which were conventionally divided or loosely coupled in the past, combine their methodologies and merge together to provide new analytical and computational tools. This is especially evident in the area of multibody system dynamics, a branch of computational mechanics dealing with modelling principles and numerical methods for dynamic analysis, simulation and control of mechanical systems.

Originating in analytical and continuum mechanics, as well as in computer science and applied mathematics, modelling methodologies and computational procedures of multibody system dynamics provide a basis for dynamic analysis and virtual prototyping of innovative applications in many fields of contemporary engineering. With the utilization of the computational models and algorithms that classically belonged to different fields of applied science, where, in certain applications, several physical models co-exist and interact within the same simulation procedure, multibody system dynamics delivers reliable simulation platforms for diverse highly-developed industrial products, such as vehicle and railway systems, aeronautical and space vehicles, robotic and autonomous platforms, biomechanical applications and nano-technologies.

However, since application-based modelling and successful implementation of computational methodologies raise many questions in terms of new solutions and optimal use of specific models and numerical procedures, multibody system dynamics is a very active research field. To maintain this development and provide a platform to discuss relevant scientific topics in this rapidly growing discipline, the 2013 edition of the ECCOMAS Thematic Conference on Multibody Dynamics was held in Zagreb, Croatia, and organized at the University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, from 1 to 4 July 2013. More than 250 participants from 38 countries participated in the event.

This book is based on the revised and extended versions of the papers presented at the conference, reporting on the state-of-the-art in the advances of computational multibody dynamics, from the recent theoretical developments to practical engineering applications. Besides ‘traditional’ multibody topics, such as terrestrial vehicles dynamics and robotical systems, as well as applications in aerospace and (today very relevant) wind turbine modelling, certain chapters of the book also reflect new frontiers in the domain of multibody system dynamics, from coupled

problems with the fluid domain and biomolecular applications to geometric integrators and variational formulations. Such a broad spectrum of topics demonstrates the vitality of this branch of computational mechanics, the roots of which can be traced far back in the history of modern engineering (as it is also documented in one of the contributions), but today, it plays a central role in the numerical modelling and optimisation of mechanical systems in a wide range of areas of scientific and engineering relevance.

The book is primarily intended for experienced researchers and doctoral students who are familiar with the fundamentals and wish to study or advance the state of the art on a particular topic in the field of multibody system dynamics. Nevertheless, practicing engineers could also benefit from it, as a variety of the presented applications show strong potential of the multibody modelling concepts, which can serve as an inspiration for further original contributions in engineering and related disciplines, such as bioengineering and applied physics. Furthermore, these pages will inform researchers active in the modern multibody dynamics about some of the principal research directions and recent achievements, as well as of the state-of-the-art applications by some of the most active researchers and prominent experts in the field.

I am grateful to all contributing authors for their active participation and for the time and effort they devoted to the completion of their contributions. I am also very much indebted to the members of the Scientific Committee for their valuable suggestions and support in the organisation of the conference, as this book is its outgrowth. The event was supported by a number of distinguished international institutions, such as ASME, IUTAM, IFToMM, and, of course, ECCOMAS, and the personal involvement of the colleagues who contributed to this support is very much appreciated.

Last but not least, I would like to thank my collaborators at the University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, who have participated in many activities during the recent years and, directly or indirectly, contributed to the publication of this book.

Zagreb, March 2014 Zdravko Terze

Contents

Chapter 1
Sensitivity Analysis of Multibody Dynamic Systems Modeled by ODEs and DAEs

Daniel Dopico, Adrian Sandu, Corina Sandu and Yitao Zhu

Abstract The optimization of the dynamic response of multibody dynamic systems is a complex and open problem. It relies on using the equations of motion of the system, which amplifies the level of complexity of the problem substantially, compared to other types of optimization. In the context of this kind of optimization, the sensitivity analysis of the dynamic response of the system is a key element. Two main techniques are currently available for the sensitivity analysis of the response of a dynamical system: the direct differentiation method and the adjoint variable method. In this work, different formulations of the equations of motion with dependent coordinates are employed and their sensitivity equations obtained. Direct and adjoint sensitivities are convenient for different types of problems but both methods require accurate derivatives of the equations of motion considered. Both approaches are employed in this work; the direct and adjoint sensitivity equations are obtained for index-3 differential-algebraic equations (DAE), index-1 DAE, and penalty formulations. The adjoint sensitivity for the penalty formulations introduced here is completely novel.

1.1 Introduction

During the last few decades, the multibody dynamics community dedicated vast amounts of time and effort to develop accurate, stable, and efficient multibody dynamics formulations, in order to handle different kind of problems. That work

D. Dopico (✉) · A. Sandu · Y. Zhu
Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, USA
e-mail: ddopico@vt.edu

D. Dopico · C. Sandu · Y. Zhu
Advanced Vehicle Dynamics Lab, Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, USA

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_1,

led to the development of several different families of methods able to solve the forward dynamics of multibody systems [12, 15]. Nowadays, more and more types of problems and phenomena are being taken into account in simulations.

One of the most interesting problems, which brought attraction since the early developments of the multibody systems techniques, is the optimization of the dynamic response of mechanical systems [13]. Currently, with the improvement of computer technology, computational power, as well as various modeling and analysis methods, the possibility of performing the sensitivity analysis and optimization of complex industrial problems is among the top priorities, and it is still an open research topic in the multibody dynamics community.

In general, the equations of motion (EOM) of multibody systems constitute an index-3 system of differential-algebraic equations (index-3 DAE system) which is usually not solved directly because of the numerical difficulties involved [1, 2]. Alternative formulations of the EOM include one or more of the following techniques: index reduction, stabilization, penalty and augmented Lagrangian techniques, nullspace or coordinate partitioning methods, and projections onto the constraints manifolds. There are two main techniques employed to calculate the sensitivity equations of multibody systems: the direct differentiation method and the adjoint variable method. Direct and adjoint sensitivity are convenient for different types of problems and both of them involve similar terms derived from the EOM. Many derivatives can be obtained analytically and implemented as general sensitivity formulations. Prior work on DAE adjoints for multibody dynamics [9, 21, 22] has been thoroughly reviewed at the beginning of this study.

In this work we bring new contributions to the state-of-the-art in sensitivity analysis for multibody dynamic systems by developing the direct and adjoint sensitivity equations for index-3 differential-algebraic equations (DAE), index-1 DAE, and penalty formulations.

1.2 Equations of Motion

All the formulations of the equations of motion considered in this work use dependent coordinates, for sensitivity analysis of formulations using independent coordinates (degrees of freedom) the reader is referred to [10].

1.2.1 Equations of Motion: Index-3 DAE Formulation

The equations of motion of a multibody system written in dependent coordinates constitute an index-3 DAE system. Assume that the configuration of a multibody system is given by a set of n coordinates $\mathbf{q} \in \mathbb{R}^n$, related by a set of m holonomic constraint equations

$$\boldsymbol{\Phi}(t, \mathbf{q}, \boldsymbol{\rho}) = \mathbf{0} \in \mathbb{R}^m, \tag{1.1}$$

where $\boldsymbol{\rho} \in \mathbb{R}^p$ is a vector of parameters of the system. Some parameter may describe the geometry of the system and therefore it may affect the constraints (1.1).

The constraint equations (1.1) allow to obtain several kinematic relations. Any virtual displacements of the system coordinates $\delta\mathbf{q}^*$, with the time held fixed, have to satisfy the following equations:

$$\boldsymbol{\Phi}_\mathbf{q}\, \delta\mathbf{q}^* = \mathbf{0}. \tag{1.2a}$$

The velocities and accelerations of the system have to fulfill the following equations

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_\mathbf{q}\dot{\mathbf{q}} + \boldsymbol{\Phi}_t = \mathbf{0} \Rightarrow \boldsymbol{\Phi}_\mathbf{q}\dot{\mathbf{q}} = -\boldsymbol{\Phi}_t = \mathbf{b}, \tag{1.2b}$$

$$\ddot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_\mathbf{q}\ddot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_\mathbf{q}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_t = \mathbf{0} \Rightarrow \boldsymbol{\Phi}_\mathbf{q}\ddot{\mathbf{q}} = -\dot{\boldsymbol{\Phi}}_\mathbf{q}\dot{\mathbf{q}} - \dot{\boldsymbol{\Phi}}_t = \mathbf{c}. \tag{1.2c}$$

The equations of motion constitute an index-3 system of $n + m$ differential-algebraic equations (DAEs)

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\lambda} = \mathbf{Q}, \tag{1.3a}$$

$$\boldsymbol{\Phi} = \mathbf{0}, \tag{1.3b}$$

where $\mathbf{M} = \mathbf{M}(\mathbf{q}, \boldsymbol{\rho}) \in \mathbb{R}^{n\times n}$ is the mass matrix, $\mathbf{Q} = \mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\rho}) \in \mathbb{R}^n$ contains the generalized forces and may also include the Coriolis and centrifugal effects (if the formulation needs them), $\boldsymbol{\Phi}_\mathbf{q} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of the constraints (1.3b), and $\boldsymbol{\lambda} \in \mathbb{R}^m$ are the Lagrange multipliers associated with the constraints.

1.2.1.1 Direct Integration of the Index-3 Equations of Motion

The direct numerical integration of the index-3 DAE equations of motion poses a number of numerical difficulties, including ill-conditioning for small time steps and instability problems that make the direct solution of the equations not recommendable.

The instability problems come from the fact that only the position level constraints are considered in the formulation and this causes an unstable behavior when integrated with the classical time-stepping schemes. The explanation for it is that the equations of motion only impose the satisfaction of the constraints themselves, but no integrator can automatically guarantee that the velocities and the accelerations will remain onto their respective manifolds of the constraints derivatives since they are weak invariants of the equations of motion [16]. To avoid this problem the authors recommend to combine the direct integration of Eqs. (1.3a), (1.3b) with the use of projection techniques like proposed in [7] or to reformulate the problem as an Augmented Lagrangian approach with projections like in [8], to solve the forward dynamics.

The ill-conditioning problem was also addressed in the past by several authors and it will be presented in this section particularized for the single-step implicit trapezoidal rule, an integrator belonging to the Newmark family; the extension to other

implicit integrators (in particular to general Newmark integrators) is straightforward. The problem posed here will be illustrative later on, since the sensitivity equations developed later in this work will inherit the same issue. Let

$$\mathbf{f} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q} \\ \boldsymbol{\Phi} \end{bmatrix} \tag{1.4}$$

be the residual of the equations of motion (1.3), meaning that Eq. (1.4) have to be equal to zero.

Using an implicit integrator (e.g. trapezoidal rule) to integrate the previous equations as described in [7]

$$\dot{\mathbf{q}}_{n+1} = \frac{2}{h}\mathbf{q}_{n+1} + \hat{\dot{\mathbf{q}}}_n; \quad \hat{\dot{\mathbf{q}}}_n = -\left(\frac{2}{h}\mathbf{q}_n + \dot{\mathbf{q}}_n\right) \tag{1.5a}$$

$$\ddot{\mathbf{q}}_{n+1} = \frac{4}{h^2}\mathbf{q}_{n+1} + \hat{\ddot{\mathbf{q}}}_n; \quad \hat{\ddot{\mathbf{q}}}_n = -\left(\frac{4}{h^2}\mathbf{q}_n + \frac{4}{h}\dot{\mathbf{q}}_n + \ddot{\mathbf{q}}_n\right) \tag{1.5b}$$

where n is the time step index and h the time step.

Replacing Eqs. (1.5a), (1.5b) in (1.4), a nonlinear system of algebraic equations in $n+1$ is obtained that can be solved using following Newton iteration

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{(i)} \Delta\mathbf{y}_{n+1}^{(i+1)} = -\mathbf{f}^{(i)} \tag{1.6a}$$

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right] = \begin{bmatrix} \frac{4}{h^2}\mathbf{M} + \frac{2}{h}\mathbf{C} + \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \tag{1.6b}$$

where $\mathbf{y} = \begin{bmatrix} \mathbf{q} & \boldsymbol{\lambda} \end{bmatrix}^{\mathrm{T}}$.

The solution of Eq. (1.6a) for small time steps poses severe issues: it was reported in [3] that the propagation of errors in the solution of the Lagrange multipliers is of order $\mathcal{O}\left(h^{-2}\right)$ and the condition number of the tangent matrix (1.6b) is of order $\mathcal{O}\left(h^{-4}\right)$ (ill conditioned for small time steps). Several authors proposed the scaling of the equations to alleviate these problems [3, 4, 19].

In [3] a specific scaling for Eq. (1.6a) with Newmark integrators is proposed, which can be easily particularized for the particular case of the trapezoidal rule by scaling the first n equations in the residual (1.4) and the Lagrange multipliers by a factor of $h^2/4$, leading to the following scaled equations and states

$$\left[\frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{y}}}\right]^{(i)} \Delta\bar{\mathbf{y}}_{n+1}^{(i+1)} = -\bar{\mathbf{f}}^{(i)} \tag{1.7a}$$

$$\bar{\mathbf{f}} = \begin{bmatrix} \frac{h^2}{4}\left(\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}\right) \\ \boldsymbol{\Phi} \end{bmatrix} \tag{1.7b}$$

$$\begin{bmatrix} \frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{y}}} \end{bmatrix} = \begin{bmatrix} \mathbf{M} + \frac{h}{2}\mathbf{C} + \frac{h^2}{4}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \tag{1.7c}$$

where $\bar{\mathbf{y}} = \begin{bmatrix} \mathbf{q} & \bar{\boldsymbol{\lambda}} \end{bmatrix}^{\mathrm{T}}$ are the scaled states and $\bar{\boldsymbol{\lambda}} = \left(h^2/4\right)\boldsymbol{\lambda}$ are the scaled Lagrange multipliers.

The suggested scaling leads both the propagation of errors in the solution of the Lagrange multipliers and the condition number of the tangent matrix (1.6b) to order $\mathcal{O}\left(h^0\right)$.

1.2.2 Equations of Motion: Index-1 DAE Formulation

The difficulties mentioned before to numerically solve Eq. (1.3) make a good case for reformulating the problem to formulations that are easier to solve. Index reduction is a common technique [1]. Differentiating (1.3b) twice leads to the following index-1 DAE system:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q}, \tag{1.8a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} - \boldsymbol{\Phi}_t = \mathbf{c}. \tag{1.8b}$$

A direct numerical solution of (1.8a), (1.8b) suffers from drift-off [11], meaning that any small perturbation in the acceleration constraints leads to an error in the position constraints that grows quadratically with time: $\ddot{\boldsymbol{\Phi}} = \epsilon_1 \Rightarrow \dot{\boldsymbol{\Phi}} = \epsilon_1 t + \epsilon_2 \Rightarrow \boldsymbol{\Phi} = \epsilon_1 t^2/2 + \epsilon_2 t + \epsilon_3$.

1.2.3 Equations of Motion: Penalty Formulation

In [5] the penalty formulation of multibody dynamics is introduced. The equations constitute a system of ordinary differential equations (ODEs) of dimension n:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\left(\ddot{\boldsymbol{\Phi}} + 2\xi\omega\dot{\boldsymbol{\Phi}} + \omega^2\boldsymbol{\Phi}\right) = \mathbf{Q} \tag{1.9}$$

where $\boldsymbol{\alpha}$ is the penalty matrix and ξ, ω are coefficients of the method. An equivalent formulation is

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} = \bar{\mathbf{Q}}, \tag{1.10a}$$

$$\bar{\mathbf{M}} \equiv \mathbf{M} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}, \tag{1.10b}$$

$$\bar{\mathbf{Q}} \equiv \mathbf{Q} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha} \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}} \dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_t + 2\xi\omega \dot{\boldsymbol{\Phi}} + \omega^2 \boldsymbol{\Phi} \right). \tag{1.10c}$$

Comparing (1.9) and (1.3a) it can be seen that the penalty equations approximate the Lagrange multipliers with the following term

$$\boldsymbol{\lambda}^* = \boldsymbol{\alpha} \left(\ddot{\boldsymbol{\Phi}} + 2\xi\omega \dot{\boldsymbol{\Phi}} + \omega^2 \boldsymbol{\Phi} \right), \tag{1.11}$$

where the star means approximated or fictitious Lagrange multipliers, as opposed to the real ones which arise in DAE formulations.

Assuming that $\boldsymbol{\alpha}$ is a diagonal matrix with the penalty factors for each constraint on the diagonal, each entry of $\boldsymbol{\lambda}^*$ in (1.11) is the equation of a one degree of freedom oscillatory system. The coefficients of the oscillatory system are usually selected as $\omega = 10$ and $\xi = 1$, which corresponds to a critically damped system.

The penalty equations (1.9) are equivalent to the original DAE for infinite penalty factors. In floating point computing, loss-of-significance errors appear in the first parenthesis of Eq. (1.10a) for large penalties, thus, the selection of the penalty factors is a sensitive issue that the analyst has to solve. In practice this formulation does not satisfy exactly either constraint equation ($\boldsymbol{\Phi} = \mathbf{0}$, $\dot{\boldsymbol{\Phi}} = \mathbf{0}$, or $\ddot{\boldsymbol{\Phi}} = \mathbf{0}$) but it approximately satisfies the equation $\ddot{\boldsymbol{\Phi}} + 2\xi\omega \dot{\boldsymbol{\Phi}} + \omega^2 \boldsymbol{\Phi} = \mathbf{0}$.

1.3 Direct Sensitivity Analysis

In this section the sensitivity equations for the equations of motion presented in Sect. 1.2 are derived based on the direct sensitivity method.

Consider the case where the equations of motion (EOM) dependent on the vector of parameters $\boldsymbol{\rho} \in \mathbb{R}^p$. The following objective function is defined in terms of the parameters, on the states $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$, and on the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^m$

$$\psi = w \left(\mathbf{q}_F, \dot{\mathbf{q}}_F, \ddot{\mathbf{q}}_F, \boldsymbol{\rho}_F, \boldsymbol{\lambda}_F \right) + \int_{t_0}^{t_F} g \left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\rho} \right) \mathrm{d}t. \tag{1.12}$$

where the subindex F means evaluation at the final time t_F.

The sensitivity analysis techniques discussed herein will evaluate the gradient of the objective function with respect to parameters:

$$\nabla_{\boldsymbol{\rho}} \psi = \left(d\psi / d\boldsymbol{\rho} \right)^{\mathrm{T}}. \tag{1.13}$$

1.3.1 Direct Sensitivity: Index-3 Formulation

The direct sensitivity method for the sensitivity analysis using the index-3 formulation in Sect. 1.2.1 was developed in [14] for objective functions dependent on

$\mathbf{q}$, $\dot{\mathbf{q}}$, $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$ and variable time limits t_F. In this section the results are revisited also for objective functions depending on $\ddot{\mathbf{q}}$ but fixed end time t_F.

Differentiating (1.12)

$$\nabla_{\boldsymbol{\rho}}\psi^{\mathrm{T}} = \left(w_{\mathbf{q}}\mathbf{q}_{\boldsymbol{\rho}} + w_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_{\boldsymbol{\rho}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}} + w_{\boldsymbol{\rho}}\right)_F + \int_{t_0}^{t_F} \left(g_{\mathbf{q}}\mathbf{q}_{\boldsymbol{\rho}} + g_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_{\boldsymbol{\rho}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}} + g_{\boldsymbol{\rho}}\right) \mathrm{d}t. \tag{1.14}$$

In Eq. (1.14) the derivatives of functions w and g are known, since the objective function has a known expression. The derivatives $\mathbf{q}_{\boldsymbol{\rho}}$, $\dot{\mathbf{q}}_{\boldsymbol{\rho}}$, $\ddot{\mathbf{q}}_{\boldsymbol{\rho}}$ and $\boldsymbol{\lambda}_{\boldsymbol{\rho}}$ are the sensitivities of the solution of the dynamical system. These can be obtained by differentiating (1.3) with respect to each one of the parameters:

$$\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}\rho_k}\ddot{\mathbf{q}} + \mathbf{M}\frac{\partial\ddot{\mathbf{q}}}{\partial\rho_k} + \frac{\mathrm{d}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}}{\mathrm{d}\rho_k}\boldsymbol{\lambda} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\frac{\partial\boldsymbol{\lambda}}{\partial\rho_k} = \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\rho_k}, \tag{1.15}$$

$$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}\rho_k} = \mathbf{0}, \quad k = 1, \ldots, m. \tag{1.16}$$

Expanding the total derivatives and grouping them together in matrix notation, leads to the following set of p DAEs, each one of them called a Tangent Linear Model (TLM):

$$\mathbf{M}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + \mathbf{C}\dot{\mathbf{q}}_{\boldsymbol{\rho}} + \left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right)\mathbf{q}_{\boldsymbol{\rho}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\boldsymbol{\rho}} = \mathbf{Q}_{\boldsymbol{\rho}} - \mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda}, \tag{1.17a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{q}_{\boldsymbol{\rho}} = -\boldsymbol{\Phi}_{\boldsymbol{\rho}}, \tag{1.17b}$$

where $\mathbf{K} = -\mathbf{Q}_{\mathbf{q}}$, $\mathbf{C} = -\mathbf{Q}_{\dot{\mathbf{q}}}$, and the following terms are tensor-vector products: $\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} \equiv \mathbf{M}_{\mathbf{q}} \otimes \ddot{\mathbf{q}}$, $\boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} \equiv \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}} \otimes \boldsymbol{\lambda}$, $\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} \equiv \mathbf{M}_{\boldsymbol{\rho}} \otimes \ddot{\mathbf{q}}$, $\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} \equiv \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}} \otimes \boldsymbol{\lambda}$.

The TLM (1.17a), (1.17b) needs for the following $2np$ initial conditions

$$\mathbf{q}_{\boldsymbol{\rho}}(t_0) = \mathbf{q}_{\boldsymbol{\rho}0}, \tag{1.18a}$$

$$\dot{\mathbf{q}}_{\boldsymbol{\rho}}(t_0) = \dot{\mathbf{q}}_{\boldsymbol{\rho}0}. \tag{1.18b}$$

The initial conditions (1.18) are not independent, since they have to satisfy the following constraint equations

$$\frac{\mathrm{d}\boldsymbol{\Phi}(t_0)}{\mathrm{d}\boldsymbol{\rho}} = \mathbf{0} \rightarrow \left[\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{q}_{\boldsymbol{\rho}}\right]_0 = -\boldsymbol{\Phi}_{\boldsymbol{\rho}0}, \tag{1.19a}$$

$$\frac{\mathrm{d}\dot{\boldsymbol{\Phi}}(t_0)}{\mathrm{d}\boldsymbol{\rho}} = \mathbf{0} \rightarrow \left[\boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}_{\boldsymbol{\rho}}\right]_0 = -\left[\left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}}\right)\mathbf{q}_{\boldsymbol{\rho}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\boldsymbol{\rho}}\right]_0 \tag{1.19b}$$

where the sub-index 0 means evaluation at the initial time t_0, as indicated in the Appendix.

Consequently $n - \text{rank}\left(\mathbf{\Phi_q}\right)$ independent sensitivities can be chosen from (1.19a) and $n - \text{rank}\left(\mathbf{\Phi_q}\right)$ independent "velocity" sensitivities from (1.19b). That means that the impact of the parameters on the initial configuration of the system, by means of a subset of degrees of freedom, can be decided as an input to the problem.

1.3.1.1 Direct Integration of the Index-3 TLM

The tangent linear model DAE in Eqs. (1.17a), (1.17b) can be directly integrated in the same way that the equations of motion in Sect. 1.2.1.1. The trapezoidal rule equations for the sensitivities

$$\dot{\mathbf{q}}_\rho^{n+1} = \frac{2}{h}\mathbf{q}_\rho^{n+1} + \hat{\dot{\mathbf{q}}}_\rho^{n}; \quad \hat{\dot{\mathbf{q}}}_\rho^{n} = -\left(\frac{2}{h}\mathbf{q}_\rho^{n} + \dot{\mathbf{q}}_\rho^{n}\right) \tag{1.20a}$$

$$\ddot{\mathbf{q}}_\rho^{n+1} = \frac{4}{h^2}\mathbf{q}_\rho^{n+1} + \hat{\ddot{\mathbf{q}}}_\rho^{n}; \quad \hat{\ddot{\mathbf{q}}}_\rho^{n} = -\left(\frac{4}{h^2}\mathbf{q}_\rho^{n} + \frac{4}{h}\dot{\mathbf{q}}_\rho^{n} + \ddot{\mathbf{q}}_\rho^{n}\right) \tag{1.20b}$$

where the superindex n means time step.

Replacing the integrator equations in (1.17a), (1.17b) and solving for $\mathbf{q}_\rho^{n+1}$ and $\boldsymbol{\lambda}_\rho^{n+1}$

$$\begin{bmatrix} \frac{4}{h^2}\mathbf{M} + \frac{2}{h}\mathbf{C} + \mathbf{M_q}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} & \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\rho^{n+1} \\ \boldsymbol{\lambda}_\rho^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_\rho - \mathbf{M}_\rho\ddot{\mathbf{q}} - \mathbf{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{M}\hat{\ddot{\mathbf{q}}}_\rho^{n} - \mathbf{C}\hat{\dot{\mathbf{q}}}_\rho^{n} \\ -\mathbf{\Phi}_\rho \end{bmatrix} \tag{1.21}$$

Observe that the leading matrix in Eq. (1.21) is identical to the tangent matrix (1.6b) and therefore exactly the same scaling applies here

$$\begin{bmatrix} \mathbf{M} + \frac{h}{2}\mathbf{C} + \frac{h^2}{4}\left(\mathbf{M_q}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) & \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q}_\rho^{n+1} \\ \bar{\boldsymbol{\lambda}}_\rho^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{h^2}{4}\left(\mathbf{Q}_\rho - \mathbf{M}_\rho\ddot{\mathbf{q}} - \mathbf{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{M}\hat{\ddot{\mathbf{q}}}_\rho^{n} - \mathbf{C}\hat{\dot{\mathbf{q}}}_\rho^{n}\right) \\ -\mathbf{\Phi}_\rho \end{bmatrix} \tag{1.22}$$

where $\bar{\boldsymbol{\lambda}}_\rho = \left(h^2/4\right)\boldsymbol{\lambda}_\rho$ are the Lagrange multipliers scaled sensitivities. To initialize the sensitivities $\mathbf{q}_\rho^0$ and $\dot{\mathbf{q}}_\rho^0$, Eqs. (1.19a), (1.19b) should be used. For the initialization

of $\ddot{\mathbf{q}}_\rho^0$ the index-1 TLM equations developed in the next section can be solved for $\ddot{\mathbf{q}}_\rho^0$ and $\boldsymbol{\lambda}_\rho^0$.

Observe that the systems (1.22) are cheap to solve, since they don't need to be iterated and the previous factorization of the leading matrix, from the dynamics, can be employed to solve.

1.3.2 Direct Sensitivity: Index-1 Formulation

The direct sensitivity method is now applied to the index-1 formulation discussed in Sect. 1.2. The gradient of the objective function is given again by (1.14) and the derivatives $\mathbf{q}_\rho$, $\dot{\mathbf{q}}_\rho$, $\ddot{\mathbf{q}}_\rho$ and $\boldsymbol{\lambda}_\rho$ are the sensitivities of the solution of the dynamical equations (1.8a), (1.8b). These sensitivities are obtained differentiating (1.8a), (1.8b) with respect to each one of the parameters as follows:

$$\frac{\mathrm{d}\mathbf{M}}{\mathrm{d}\rho_k}\ddot{\mathbf{q}} + \mathbf{M}\frac{\partial \ddot{\mathbf{q}}}{\partial \rho_k} + \frac{\mathrm{d}\boldsymbol{\Phi}_\mathbf{q}^\mathrm{T}}{\mathrm{d}\rho_k}\boldsymbol{\lambda} + \boldsymbol{\Phi}_\mathbf{q}^\mathrm{T}\frac{\partial \boldsymbol{\lambda}}{\partial \rho_k} = \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}\rho_k}, \tag{1.23a}$$

$$\frac{\mathrm{d}\ddot{\boldsymbol{\Phi}}}{\mathrm{d}\rho_k} = \mathbf{0}, \quad k = 1, \ldots, p. \tag{1.23b}$$

Expanding the total derivatives and grouping them together in matrix notation, leads to the following set of p DAEs, each one of them called a Tangent Linear Model (TLM):

$$\mathbf{M}\ddot{\mathbf{q}}_\rho + \mathbf{C}\dot{\mathbf{q}}_\rho + \left(\mathbf{M}_\mathbf{q}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_\mathbf{qq}^\mathrm{T}\boldsymbol{\lambda} + \mathbf{K}\right)\mathbf{q}_\rho + \boldsymbol{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\lambda}_\rho = \mathbf{Q}_\rho - \mathbf{M}_\rho\ddot{\mathbf{q}} - \boldsymbol{\Phi}_\mathbf{q\rho}^\mathrm{T}\boldsymbol{\lambda}, \tag{1.24a}$$

$$\boldsymbol{\Phi}_\mathbf{q}\ddot{\mathbf{q}}_\rho - \mathbf{c}_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_\rho + \left(\boldsymbol{\Phi}_\mathbf{qq}\ddot{\mathbf{q}} - \mathbf{c}_\mathbf{q}\right)\mathbf{q}_\rho = \mathbf{c}_\rho - \boldsymbol{\Phi}_\mathbf{q\rho}\ddot{\mathbf{q}} \tag{1.24b}$$

where

$$\mathbf{c}_\mathbf{q} = -\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_\mathbf{q}\dot{\mathbf{q}} - \left(\dot{\boldsymbol{\Phi}}_t\right)_\mathbf{q}, \tag{1.24c}$$

$$\mathbf{c}_{\dot{\mathbf{q}}} = -\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_{\dot{\mathbf{q}}}\dot{\mathbf{q}} - \dot{\boldsymbol{\Phi}}_\mathbf{q} - \left(\dot{\boldsymbol{\Phi}}_t\right)_{\dot{\mathbf{q}}} = -\boldsymbol{\Phi}_\mathbf{qq}\dot{\mathbf{q}} - \dot{\boldsymbol{\Phi}}_\mathbf{q} - \boldsymbol{\Phi}_{t\mathbf{q}}, \tag{1.24d}$$

$$\mathbf{c}_\rho = -\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_\rho\dot{\mathbf{q}} - \left(\dot{\boldsymbol{\Phi}}_t\right)_\rho. \tag{1.24e}$$

The identities $\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_{\dot{\mathbf{q}}} = \boldsymbol{\Phi}_\mathbf{qq}$ and $\left(\dot{\boldsymbol{\Phi}}_t\right)_{\dot{\mathbf{q}}} = \boldsymbol{\Phi}_{t\mathbf{q}}$ were used, and the tensor-vector product rules, with the operator $\otimes$, were applied to the terms involving $\boldsymbol{\Phi}_\mathbf{qq}$, $\boldsymbol{\Phi}_\mathbf{q\rho}$, $\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_\mathbf{q}$ and $\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\right)_\rho$.

The approach presented for the index-3 TLM can be employed here to obtain the initial conditions of the index-1 TLM [Eqs. (1.19a), (1.19b)]. On the other hand the approach presented here, can be employed to initialize the sensitivities $\ddot{\mathbf{q}}_\rho$

(if necessary) for both the index-1 and the index-3 TLM. From Eqs. (1.24a), (1.24b), (1.24c), (1.24d), (1.24e)

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\rho} \\ \boldsymbol{\lambda}_{\rho} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{\rho} - \mathbf{M}_{\rho}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{C}\dot{\mathbf{q}}_{\rho} - \left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right)\mathbf{q}_{\rho} \\ \mathbf{c}_{\rho} - \boldsymbol{\Phi}_{\mathbf{q}\rho}\ddot{\mathbf{q}} + \mathbf{c}_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_{\rho} - \left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} + \mathbf{c}_{\mathbf{q}}\right)\mathbf{q}_{\rho} \end{bmatrix} \tag{1.25}$$

1.3.3 Direct Sensitivity: Penalty Formulation

The direct sensitivity method for the sensitivity analysis using the penalty formulation was initially developed in [17, 18]. We seek to obtain the sensitivities of the objective function

$$\psi = w\left(\mathbf{q}_F, \dot{\mathbf{q}}_F, \ddot{\mathbf{q}}_F, \boldsymbol{\rho}_F, \boldsymbol{\lambda}_F^*\right) + \int_{t_0}^{t_F} g\left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\rho}, \boldsymbol{\lambda}^*\right) \mathrm{d}t. \tag{1.26}$$

The Lagrange multipliers $\boldsymbol{\lambda}$ in (1.12) are replaced by the approximate Lagrange multipliers $\boldsymbol{\lambda}^*$ in (1.26). These variables don't show up in the penalty equations of motion (1.9) and they have to be approximated by (1.11). The initial and final times are considered here independent of the parameters, but more general objective functions are possible.

The gradient of the objective function (1.26) can be obtained by the following expressions

$$\begin{aligned} \nabla_{\boldsymbol{\rho}}\psi^{\mathrm{T}} = \frac{\mathrm{d}\psi}{\mathrm{d}\boldsymbol{\rho}} = &\left[w_{\mathbf{q}}\mathbf{q}_{\rho} + w_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_{\rho} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + w_{\rho} + w_{\boldsymbol{\lambda}^*}\alpha\left(\frac{\mathrm{d}\ddot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} + 2\xi\omega\frac{\mathrm{d}\dot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} + \omega^2\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}\boldsymbol{\rho}}\right)\right]_F \\ &+ \int_{t_0}^{t_F} \left[g_{\mathbf{q}}\mathbf{q}_{\rho} + g_{\dot{\mathbf{q}}}\dot{\mathbf{q}}_{\rho} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + g_{\rho} + g_{\boldsymbol{\lambda}^*}\alpha\left(\frac{\mathrm{d}\ddot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} + 2\xi\omega\frac{\mathrm{d}\dot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} + \omega^2\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}\boldsymbol{\rho}}\right)\right] \mathrm{d}t, \end{aligned} \tag{1.27a}$$

$$\begin{aligned} \frac{\mathrm{d}\ddot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} = &\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}}_{\rho} + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}}\right)\dot{\mathbf{q}}_{\rho} + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t}\right)_{\mathbf{q}}\right)\mathbf{q}_{\rho} \\ &+ \boldsymbol{\Phi}_{\mathbf{q}\rho}\ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)_{\rho}\dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t}\right)_{\rho}, \end{aligned} \tag{1.27b}$$

$$\frac{\mathrm{d}\dot{\boldsymbol{\Phi}}}{\mathrm{d}\boldsymbol{\rho}} = \boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}_{\rho} + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}}\right)\mathbf{q}_{\rho} + \boldsymbol{\Phi}_{\mathbf{q}\rho}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\rho}, \tag{1.27c}$$

$$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}\boldsymbol{\rho}} = \boldsymbol{\Phi}_{\mathbf{q}}\mathbf{q}_{\rho} + \boldsymbol{\Phi}_{\rho}, \tag{1.27d}$$

where the identities $\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\right)_{\dot{\mathbf{q}}} = \mathbf{\Phi}_{\mathbf{qq}}$ and $\left(\dot{\mathbf{\Phi}}_{t}\right)_{\dot{\mathbf{q}}} = \mathbf{\Phi}_{t\mathbf{q}}$ were used, and the tensor-vector product rules mentioned in the Appendix, with the operator $\otimes$, were applied to the terms involving $\mathbf{\Phi}_{\mathbf{qq}}$, $\mathbf{\Phi}_{\mathbf{q}\rho}$, $\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\right)_{\mathbf{q}}$ and $\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\right)_{\rho}$.

In Eqs. (1.27a), (1.27b), (1.27c), (1.27d) the derivatives of functions w and g are known, since the objective function has a known expression. The derivatives $\mathbf{q}_{\rho}$, $\dot{\mathbf{q}}_{\rho}$ and $\ddot{\mathbf{q}}_{\rho}$ are the sensitivities of the solution of the dynamical system, and they are obtained by differentiating Eq. (1.9) with respect to each one of the parameters :

$$\frac{\mathrm{d}\bar{\mathbf{M}}}{\mathrm{d}\rho_k}\ddot{\mathbf{q}} + \bar{\mathbf{M}}\frac{\partial\ddot{\mathbf{q}}}{\partial\rho_k} = \frac{\mathrm{d}\bar{\mathbf{Q}}}{\mathrm{d}\rho_k}, \quad k = 1, \ldots, p. \tag{1.28}$$

Expanding the total derivatives and grouping them together in matrix notation leads to the following set of p ODEs, each one of them called a Tangent Linear Model (TLM):

$$\bar{\mathbf{M}}\ddot{\mathbf{q}}_{\rho} + \bar{\mathbf{C}}\dot{\mathbf{q}}_{\rho} + \left(\bar{\mathbf{K}} + \bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}}\right)\mathbf{q}_{\rho} = \bar{\mathbf{Q}}_{\rho} - \bar{\mathbf{M}}_{\rho}\ddot{\mathbf{q}}, \tag{1.29a}$$

$$\mathbf{q}_{\rho}\left(t_0\right) = \mathbf{q}_{\rho 0}, \tag{1.29b}$$

$$\dot{\mathbf{q}}_{\rho}\left(t_0\right) = \dot{\mathbf{q}}_{\rho 0}. \tag{1.29c}$$

Observe that each TLM (1.29a), (1.29b), (1.29c) is a system of only n equations, compared to the $n + m$ equations of the index-3 and index-1 tangent linear DAEs in Sects. 1.3.1 and 1.3.2 respectively. Equations (1.29b) and (1.29c) are the initial conditions for the tangent linear ODE and they can be obtained by a similar approach to the one in Sect. 1.3.1.

The following terms appear in (1.29a):

$$\begin{aligned}\bar{\mathbf{K}} = -\frac{\partial\bar{\mathbf{Q}}}{\partial\mathbf{q}} &= \mathbf{K} + \mathbf{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{t} + 2\xi\omega\dot{\mathbf{\Phi}} + \omega^2\mathbf{\Phi}\right)\\ &\quad + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}} + \left(\dot{\mathbf{\Phi}}_{t}\right)_{\mathbf{q}} + 2\xi\omega\left(\mathbf{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \mathbf{\Phi}_{t\mathbf{q}}\right) + \omega^2\mathbf{\Phi}_{\mathbf{q}}\right)\end{aligned} \tag{1.30a}$$

$$\bar{\mathbf{C}} = -\frac{\partial\bar{\mathbf{Q}}}{\partial\dot{\mathbf{q}}} = \mathbf{C} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\mathbf{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{\mathbf{q}} + \mathbf{\Phi}_{t\mathbf{q}} + 2\xi\omega\mathbf{\Phi}_{\mathbf{q}}\right) \tag{1.30b}$$

$$\begin{aligned}\bar{\mathbf{Q}}_{\rho} = \frac{\partial\bar{\mathbf{Q}}}{\partial\boldsymbol{\rho}} &= \mathbf{Q}_{\rho} - \mathbf{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{t} + 2\xi\omega\dot{\mathbf{\Phi}} + \omega^2\mathbf{\Phi}\right)\\ &\quad - \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\rho} + \dot{\mathbf{\Phi}}_{t\rho} + 2\xi\omega\dot{\mathbf{\Phi}}_{\rho} + \omega^2\mathbf{\Phi}_{\rho}\right)\end{aligned} \tag{1.30c}$$

$$\bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}} = \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\alpha\left(\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}}\right) + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\mathbf{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} \tag{1.30d}$$

$$\bar{\mathbf{M}}_\rho \ddot{\mathbf{q}} = \mathbf{M}_\rho \ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}} \alpha \left(\boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}} \right) + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \alpha \boldsymbol{\Phi}_{\mathbf{q}\rho} \ddot{\mathbf{q}} \tag{1.30e}$$

We have $\mathbf{K} = -\mathbf{Q}_{\mathbf{q}}$ in Eq. (1.30a) and $\mathbf{C} = -\mathbf{Q}_{\dot{\mathbf{q}}}$ in (1.30b). In Eqs. (1.30d) and (1.30e) the following tensor-vector products are calculated: $\mathbf{M}_{\mathbf{q}} \ddot{\mathbf{q}} = \mathbf{M}_{\mathbf{q}} \otimes \ddot{\mathbf{q}}$ and $\mathbf{M}_\rho \ddot{\mathbf{q}} = \mathbf{M}_\rho \otimes \ddot{\mathbf{q}}$.

1.4 Adjoint Sensitivity Analysis

In this section the sensitivity equations for the equations of motion presented in Sect. 1.2 are derived based on the adjoint variable method.

1.4.1 Adjoint Sensitivity: Index-3 Formulation

The adjoint sensitivity method for the sensitivity analysis using the index-3 formulation in Sect. 1.2 was developed in [14] for objective functions dependent on $\mathbf{q}$, $\dot{\mathbf{q}}$, $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$ and variable time limits t_F. In this section the results are revisited also for objective functions depending on $\ddot{\mathbf{q}}$ but fixed end time t_F.

Considering the equations of motion (1.3) the gradient (1.13) can be indirectly obtained via the Lagrangian function

$$\begin{aligned} \mathcal{L}(\boldsymbol{\rho}) = w\left(\mathbf{q}_F, \dot{\mathbf{q}}_F, \ddot{\mathbf{q}}_F, \boldsymbol{\rho}_F, \boldsymbol{\lambda}_F\right) + \int_{t_0}^{t_F} g\left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\rho}\right) \mathrm{d}t \\ - \int_{t_0}^{t_F} \boldsymbol{\mu}^{\mathrm{T}} \left(\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\lambda} - \mathbf{Q} \right) \mathrm{d}t - \int_{t_0}^{t_F} \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{\Phi} \mathrm{d}t \end{aligned} \tag{1.31}$$

and the following identity which holds along any solution of the equations of motion:

$$\nabla_{\boldsymbol{\rho}} \psi = \nabla_{\boldsymbol{\rho}} \mathcal{L}. \tag{1.32}$$

Infinitesimal variations of $\mathcal{L}$ under infinitesimal variations $\delta\boldsymbol{\rho}$ are as follows (note that the computation of $\delta\boldsymbol{\mu}$ and $\delta\boldsymbol{\lambda}$ is not needed):

$$\begin{aligned} \delta\mathcal{L} = \left[w_{\mathbf{q}} \delta\mathbf{q} + w_{\dot{\mathbf{q}}} \delta\dot{\mathbf{q}} + w_{\ddot{\mathbf{q}}} \delta\ddot{\mathbf{q}} + w_{\boldsymbol{\lambda}} \delta\boldsymbol{\lambda} + w_{\boldsymbol{\rho}} \delta\boldsymbol{\rho} \right]_F \\ + \int_{t_0}^{t_F} \left(g_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}} \left(\mathbf{M}_{\mathbf{q}} \ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}} \boldsymbol{\lambda} - \mathbf{Q}_{\mathbf{q}} \right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{\Phi}_{\mathbf{q}} \right) \delta\mathbf{q} \mathrm{d}t \\ + \int_{t_0}^{t_F} \left(g_{\dot{\mathbf{q}}} + \boldsymbol{\mu}^{\mathrm{T}} \mathbf{Q}_{\dot{\mathbf{q}}} \right) \delta\dot{\mathbf{q}} \mathrm{d}t + \int_{t_0}^{t_F} \left(g_{\ddot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}} \mathbf{M} \right) \delta\ddot{\mathbf{q}} \mathrm{d}t + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\lambda}} - \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \right) \delta\boldsymbol{\lambda} \mathrm{d}t \end{aligned} \tag{1.33}$$

$$+\int_{t_0}^{t_F}\left(g_\rho-\boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_\rho\ddot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}-\mathbf{Q}_\rho\right)-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_\rho\right)\delta\boldsymbol{\rho}\mathrm{d}t.$$

Integrating by parts the integrals involving $\delta\dot{\mathbf{q}}$, $\delta\ddot{\mathbf{q}}$:

$$\begin{aligned}\delta\mathcal{L}=&\left[w_{\mathbf{q}}\delta\mathbf{q}+w_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}}+w_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}}+w_{\boldsymbol{\lambda}}\delta\boldsymbol{\lambda}+w_{\rho}\delta\boldsymbol{\rho}\right]_F\\&+\int_{t_0}^{t_F}\left(g_{\mathbf{q}}-\boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}+\mathbf{K}\right)-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}\right)\delta\mathbf{q}\mathrm{d}t+\left(g_{\dot{\mathbf{q}}}-\boldsymbol{\mu}^{\mathrm{T}}\mathbf{C}\right)\delta\mathbf{q}\Big|_{t0}^{tF}\\&-\int_{t_0}^{t_F}\left(\frac{\mathrm{d}g_{\dot{\mathbf{q}}}}{\mathrm{d}t}-\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{C}-\boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{C}}\right)\delta\mathbf{q}\mathrm{d}t+\left(g_{\ddot{\mathbf{q}}}-\boldsymbol{\mu}^{\mathrm{T}}\mathbf{M}\right)\delta\dot{\mathbf{q}}\Big|_{t0}^{tF}\\&-\left(\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}}{\mathrm{d}t}-\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M}-\boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{M}}\right)\delta\mathbf{q}\Big|_{t0}^{tF}+\int_{t_0}^{t_F}\left(\frac{\mathrm{d}^2g_{\ddot{\mathbf{q}}}}{\mathrm{d}t^2}-\ddot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M}-2\dot{\boldsymbol{\mu}}^{\mathrm{T}}\dot{\mathbf{M}}-\boldsymbol{\mu}^{\mathrm{T}}\ddot{\mathbf{M}}\right)\delta\mathbf{q}\mathrm{d}t\\&+\int_{t_0}^{t_F}\left(g_{\boldsymbol{\lambda}}-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\right)\delta\boldsymbol{\lambda}\mathrm{d}t+\int_{t_0}^{t_F}\left(g_\rho-\boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_\rho\ddot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}-\mathbf{Q}_\rho\right)-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_\rho\right)\delta\boldsymbol{\rho}\mathrm{d}t\end{aligned}\tag{1.34}$$

When the following equations are satisfied all the integral terms that involve $\delta\mathbf{q}$ and $\delta\boldsymbol{\lambda}$ cancel:

$$\begin{aligned}&g_{\mathbf{q}}-\boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}+\mathbf{K}\right)-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}-\frac{\mathrm{d}g_{\dot{\mathbf{q}}}}{\mathrm{d}t}+\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{C}+\boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{C}}+\frac{\mathrm{d}^2g_{\ddot{\mathbf{q}}}}{\mathrm{d}t^2}\\&\quad-\ddot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M}-2\dot{\boldsymbol{\mu}}^{\mathrm{T}}\dot{\mathbf{M}}-\boldsymbol{\mu}^{\mathrm{T}}\ddot{\mathbf{M}}=\mathbf{0}\end{aligned}\tag{1.35a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu}=g_{\boldsymbol{\lambda}}^{\mathrm{T}}\tag{1.35b}$$

The initial conditions can be obtained by canceling the additional terms at the final time appearing in (1.34): $\delta\mathbf{q}_F$, $\delta\dot{\mathbf{q}}_F$, $\delta\boldsymbol{\lambda}_F$

$$\begin{aligned}&\left[w_{\mathbf{q}}+g_{\dot{\mathbf{q}}}-\boldsymbol{\mu}^{\mathrm{T}}\mathbf{C}-\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}}{\mathrm{d}t}+\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M}+\boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{M}}\right]_F\delta\mathbf{q}_F\\&\quad+\left[w_{\dot{\mathbf{q}}}+g_{\ddot{\mathbf{q}}}-\boldsymbol{\mu}^{\mathrm{T}}\mathbf{M}\right]_F\delta\dot{\mathbf{q}}_F+\left[w_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}}\right]_F+\left[w_{\boldsymbol{\lambda}}\delta\boldsymbol{\lambda}\right]_F=\mathbf{0}\end{aligned}\tag{1.35c}$$

An additional initial condition is given by (1.35b) which holds at any time, in particular at $t=t_F$.

It can be shown that the resulting adjoint system is an index-3 DAE in $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$:

$$\mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}} + \left(2\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} + \ddot{\mathbf{M}} - \dot{\mathbf{C}}\right)^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} = g_{\mathbf{q}}^{\mathrm{T}} - \frac{\mathrm{d}g_{\dot{\mathbf{q}}}}{\mathrm{d}t}^{\mathrm{T}} + \frac{\mathrm{d}^2 g_{\ddot{\mathbf{q}}}}{\mathrm{d}t^2}^{\mathrm{T}} \tag{1.36a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu} = g_{\boldsymbol{\lambda}}^{\mathrm{T}} \tag{1.36b}$$

and has the following initial conditions

$$\left[\mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu}\right]_F = \left[\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}}{\mathrm{d}t}^{\mathrm{T}} - g_{\dot{\mathbf{q}}}^{\mathrm{T}} - w_{\mathbf{q}}^{\mathrm{T}}\right]_F \tag{1.37a}$$

$$\left[\mathbf{M}^{\mathrm{T}}\boldsymbol{\mu}\right]_F = \left[g_{\ddot{\mathbf{q}}}^{\mathrm{T}} + w_{\dot{\mathbf{q}}}^{\mathrm{T}}\right]_F \tag{1.37b}$$

$$\left[w_{\ddot{\mathbf{q}}}\right]_F = \mathbf{0} \tag{1.37c}$$

$$[w_{\boldsymbol{\lambda}}|_F = \mathbf{0} \tag{1.37d}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu}\right]_F = \left[g_{\boldsymbol{\lambda}}^{\mathrm{T}}\right]_F \tag{1.37e}$$

Equations (1.37c) and (1.37d) express incompatible conditions for the objective function, meaning that the objective function cannot depend on the acceleration or Lagrange multipliers in the final time. Moreover, if $g_{\ddot{\mathbf{q}}}$, $w_{\dot{\mathbf{q}}}$ or $g_{\boldsymbol{\lambda}}$ are not equal to zero, Eqs. (1.37b) and (1.37e) constitute an incompatible system of equations in $\boldsymbol{\mu}$ which can be solved, according to [14], adding the following terms, evaluated at the final time, to the Lagrangian (1.31).

$$\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{\Phi}\left(t_F, \mathbf{q}_F, \boldsymbol{\rho}_F\right) \tag{1.38}$$

$$\boldsymbol{\eta}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}\left(t_F, \mathbf{q}_F, \dot{\mathbf{q}}_F, \boldsymbol{\rho}_F\right) \tag{1.39}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^m$ are new adjoint variables that need to be determined and the constraint equations are given in Eqs. (1.1) and (1.2b).

The addition of condition (1.38) is convenient, since the constraint equations are imposed by the index-3 formulation, nevertheless the addition of (1.39) can be problematic since they are hidden constraints not explicitly imposed by the index-3 formulation unless, for example, projection techniques are employed, as proposed in [7] and suggested in Sect. 1.2.1.1.

The addition of Eqs. (1.38) and (1.39) to the Lagrangian (1.31), contribute with the following terms to the variation (1.33)

$$\boldsymbol{\gamma}^{\mathrm{T}}\left[\boldsymbol{\Phi}_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\Phi}_{\rho}\delta\boldsymbol{\rho}\right]_F \tag{1.40}$$

$$\boldsymbol{\eta}^{\mathrm{T}}\left[\left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}}+\boldsymbol{\Phi}_{t\mathbf{q}}\right)\delta\mathbf{q}+\boldsymbol{\Phi}_{\mathbf{q}}\delta\dot{\mathbf{q}}+\left(\boldsymbol{\Phi}_{\mathbf{q}\rho}\dot{\mathbf{q}}+\boldsymbol{\Phi}_{t\rho}\right)\delta\boldsymbol{\rho}\right]_F \tag{1.41}$$

The final adjoint is not affected by these additions and Eqs. (1.36a), (1.36b) are still valid, nevertheless the initial conditions (1.37a), (1.37b), (1.37c), (1.37d), eqrefeq:indexsps3spsICs5 for the adjoint at the final time become the following

$$\left[\mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}}+\left(\dot{\mathbf{M}}-\mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\gamma}+\left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}}+\boldsymbol{\Phi}_{t\mathbf{q}}\right)^{\mathrm{T}}\boldsymbol{\eta}\right]_F=\left[\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}^{\mathrm{T}}}{\mathrm{dt}}-g_{\dot{\mathbf{q}}}^{\mathrm{T}}-w_{\mathbf{q}}^{\mathrm{T}}\right]_F \tag{1.42a}$$

$$\left[\mathbf{M}^{\mathrm{T}}\boldsymbol{\mu}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\eta}\right]_F=\left[g_{\ddot{\mathbf{q}}}^{\mathrm{T}}+w_{\dot{\mathbf{q}}}^{\mathrm{T}}\right]_F \tag{1.42b}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu}\right]_F=\left[g_{\boldsymbol{\lambda}}^{\mathrm{T}}\right]_F \tag{1.42c}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\dot{\boldsymbol{\mu}}+\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\boldsymbol{\mu}\right]_F=\left[\frac{\mathrm{d}g_{\boldsymbol{\lambda}}^{\mathrm{T}}}{\mathrm{d}t}\right]_F \tag{1.42d}$$

where Eq. (1.42d) is the derivative of (1.42c) and it was added to complete a full set of equations to determine the initial adjoint variables. The process was described by [14] and consist of solving the two sets of equations formed with (1.42b), (1.42c) to obtain $\boldsymbol{\mu}_F$ and $\boldsymbol{\eta}_F$ and (1.42a), (1.42d) to obtain $\dot{\boldsymbol{\mu}}_F$ and $\boldsymbol{\gamma}_F$.

The gradient of the objective function is obtained with the remaining terms not canceled out in the variational equation (1.34) and additional terms (1.40), (1.41):

$$\begin{aligned}\nabla_{\rho}\psi^{\mathrm{T}}=&\left[w_{\rho}+\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{\Phi}_{\rho}+\boldsymbol{\eta}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{q}\rho}\dot{\mathbf{q}}+\boldsymbol{\Phi}_{t\rho}\right)\right]_F+\left[\left(\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}}{\mathrm{dt}}-g_{\dot{\mathbf{q}}}-\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M}-\boldsymbol{\mu}^{\mathrm{T}}\left(\dot{\mathbf{M}}-\mathbf{C}\right)\right)\mathbf{q}_{\rho}\right]_0\\&-\left[\left(g_{\ddot{\mathbf{q}}}-\boldsymbol{\mu}^{\mathrm{T}}\mathbf{M}\right)\dot{\mathbf{q}}_{\rho}\right]_0+\int_{t_0}^{t_F}\left(g_{\rho}-\boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\rho}\ddot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}-\mathbf{Q}_{\rho}\right)-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\rho}\right)\mathrm{dt}\end{aligned} \tag{1.43}$$

The gradient depends on the solution of the EOM and on the adjoint differential and algebraic variables $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ which are the solution of the adjoint system. The adjoint system derived from the index-3 equations of motion is an index-3 DAE.

This formulation of the sensitivity equations is not very convenient for the case of objective functions depending on $\ddot{\mathbf{q}}$ or $\boldsymbol{\lambda}$ due to the difficulty to obtain the derivatives of these terms, involving higher order derivatives of the states or Lagrange multipliers, normally not calculated.

1.4.1.1 Direct Integration of the Adjoint DAE

The adjoint index-3 DAE (1.36a), (1.36b) can be directly integrated backward in time:

$$\mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}} + \mathbf{A}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \mathbf{B}^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} = g_{\mathbf{q}}^{\mathrm{T}} - \frac{\mathrm{d}g_{\dot{\mathbf{q}}}}{\mathrm{dt}}^{\mathrm{T}} + \frac{\mathrm{d}^2 g_{\ddot{\mathbf{q}}}}{\mathrm{dt}^2}^{\mathrm{T}} \tag{1.44a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu} = g_{\boldsymbol{\lambda}}^{\mathrm{T}} \tag{1.44b}$$

where $\mathbf{A} = 2\dot{\mathbf{M}} - \mathbf{C}$ and $\mathbf{B} = \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} + \ddot{\mathbf{M}} - \dot{\mathbf{C}}$.

The implicit trapezoidal rule equations for backward integration can be expressed as

$$\dot{\boldsymbol{\mu}}_*^n = -\frac{2}{h}\boldsymbol{\mu}_*^n + \hat{\dot{\boldsymbol{\mu}}}_*^{n+1}; \qquad \hat{\dot{\boldsymbol{\mu}}}_*^{n+1} = \frac{2}{h}\boldsymbol{\mu}_*^{n+1} - \dot{\boldsymbol{\mu}}_*^{n+1} \tag{1.45a}$$

$$\ddot{\boldsymbol{\mu}}_*^n = -\frac{4}{h^2}\boldsymbol{\mu}_*^n + \hat{\ddot{\boldsymbol{\mu}}}_*^{n+1}; \qquad \hat{\ddot{\boldsymbol{\mu}}}_*^{n+1} = -\frac{4}{h^2}\boldsymbol{\mu}_*^{n+1} + \frac{4}{h}\dot{\boldsymbol{\mu}}_*^{n+1} - \ddot{\boldsymbol{\mu}}_*^{n+1} \tag{1.45b}$$

where the subindex $*$ means that the same equations hold for $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$.

Replacing Eqs. (1.45a), (1.45b) in (1.44a), (1.44b)

$$\begin{bmatrix} \frac{4}{h^2}\mathbf{M}^{\mathrm{T}} + \frac{2}{h}\mathbf{A}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^n \\ \boldsymbol{\mu}_{\boldsymbol{\Phi}}^n \end{bmatrix} = \begin{bmatrix} g_{\mathbf{q}}^{\mathrm{T}} - \dot{g}_{\dot{\mathbf{q}}}^{\mathrm{T}} + \ddot{g}_{\ddot{\mathbf{q}}}^{\mathrm{T}} - \mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}}^{n+1} - \mathbf{A}^{\mathrm{T}}\dot{\boldsymbol{\mu}}^{n+1} \\ \mathbf{0} \end{bmatrix} \tag{1.46}$$

Observe that the system of equations (1.46) is very similar in its structure to (1.6a), except by the fact that (1.46) is linear in $\boldsymbol{\mu}^n$, $\boldsymbol{\mu}_{\boldsymbol{\Phi}}^n$ and therefore it doesn't need to be iterated. This similarity suggests the same scaling proposed before, leading to the following scaled system

$$\begin{bmatrix} \mathbf{M}^{\mathrm{T}} + \frac{h}{2}\mathbf{A}^{\mathrm{T}} + \frac{h^2}{4}\mathbf{B}^{\mathrm{T}} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^n \\ \bar{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^n \end{bmatrix} = \begin{bmatrix} \frac{h^2}{4}\left(g_{\mathbf{q}}^{\mathrm{T}} - \dot{g}_{\dot{\mathbf{q}}}^{\mathrm{T}} + \ddot{g}_{\ddot{\mathbf{q}}}^{\mathrm{T}} - \mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}}^{n+1} - \mathbf{A}^{\mathrm{T}}\dot{\boldsymbol{\mu}}^{n+1}\right) \\ \mathbf{0} \end{bmatrix} \tag{1.47}$$

where $\bar{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^n = \left(h^2/4\right)\boldsymbol{\mu}_{\boldsymbol{\Phi}}^n$ are the scaled adjoint variables associated to the constraint equations.

1.4.2 Adjoint Sensitivity: Index-1 Formulation

1.4.2.1 Approach 1

The adjoint sensitivity for the index-1 formulation was obtained in [6] for the equations of motion considered as a first order system and objective functions with variable time limits t_F but not dependent either on $\boldsymbol{\lambda}$ or $\ddot{\mathbf{q}}_F$. In this section the results are revisited also for objective functions depending on $\ddot{\mathbf{q}}_F$ and $\boldsymbol{\lambda}$ but fixed end time t_F.

Considering the equations of motion (1.8a), (1.8b), the gradient (1.13) can be obtained like in Sect. 1.4.1

$$\mathcal{L}(\boldsymbol{\rho}) = w(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\rho})_{t_F} + \int_{t_0}^{t_F} g(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\rho})\,\mathrm{dt} - \int_{t_0}^{t_F} \boldsymbol{\mu}^{\mathrm{T}} \left(\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}\right)\mathrm{dt} - \int_{t_0}^{t_F} \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}} \left(\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} - \mathbf{c}\right)\mathrm{dt} \tag{1.48}$$

Compute infinitesimal variations of $\mathcal{L}$ under infinitesimal variations $\delta\boldsymbol{\rho}$ (computation of $\delta\boldsymbol{\mu}$ and $\delta\boldsymbol{\lambda}$ not needed):

$$\begin{aligned}
\delta\mathcal{L} = {} & \left[w_{\mathbf{q}}\delta\mathbf{q} + w_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}} + w_{\boldsymbol{\lambda}}\delta\boldsymbol{\lambda} + w_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}\right]_{t_F} \\
& + \int_{t_0}^{t_F} \left(g_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} - \mathbf{c}_{\mathbf{q}}\right)\right)\delta\mathbf{q}\mathrm{dt} \\
& + \int_{t_0}^{t_F} \left(g_{\dot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{C} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}}\right)\delta\dot{\mathbf{q}}\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\ddot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{M} - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}\right)\delta\ddot{\mathbf{q}}\mathrm{dt} \\
& + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\lambda}} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\right)\delta\boldsymbol{\lambda}\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\rho}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right)\right. \\
& \left. - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)\right)\delta\boldsymbol{\rho}\mathrm{dt}
\end{aligned} \tag{1.49}$$

Integrating by parts the integrals involving $\delta\dot{\mathbf{q}}$, $\delta\ddot{\mathbf{q}}$ leads to:

$$\begin{aligned}
\delta\mathcal{L} = {} & \left[w_{\mathbf{q}}\delta\mathbf{q} + w_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}} + w_{\boldsymbol{\lambda}}\delta\boldsymbol{\lambda} + w_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}\right]_{t_F} \\
& + \int_{t_0}^{t_F} \left(g_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} - \mathbf{c}_{\mathbf{q}}\right)\right)\delta\mathbf{q}\mathrm{dt} \\
& + \left(g_{\dot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{C} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} - \int_{t_0}^{t_F} \left(\frac{\mathrm{d}g_{\dot{\mathbf{q}}}}{\mathrm{d}t} - \dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{C} - \boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{C}} + \dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{c}_{\dot{\mathbf{q}}}}{\mathrm{d}t}\right)\delta\mathbf{q}\mathrm{dt} \\
& + \left(g_{\ddot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{M} - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}\right)\delta\dot{\mathbf{q}}\Big|_{t0}^{tF} - \left(\frac{\mathrm{d}g_{\ddot{\mathbf{q}}}}{\mathrm{d}t} - \dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M} - \boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{M}} - \dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}} - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} \\
& + \int_{t_0}^{t_F} \left(\frac{\mathrm{d}^2 g_{\ddot{\mathbf{q}}}}{\mathrm{d}t^2} - \ddot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M} - 2\dot{\boldsymbol{\mu}}^{\mathrm{T}}\dot{\mathbf{M}} - \boldsymbol{\mu}^{\mathrm{T}}\ddot{\mathbf{M}} - \ddot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}} - 2\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}_{\mathbf{q}} - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\ddot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\delta\mathbf{q}\mathrm{dt} \\
& + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\lambda}} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\right)\delta\boldsymbol{\lambda}\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\rho}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)\right)\delta\boldsymbol{\rho}\mathrm{dt}
\end{aligned} \tag{1.50}$$

Canceling all the integral terms that involve $\delta\mathbf{q}$ and $\delta\boldsymbol{\lambda}$, the following adjoint DAE is obtained

$$\mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}} + \left(2\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} - \dot{\mathbf{C}} + \ddot{\mathbf{M}}\right)^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\ddot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}} + \left(\mathbf{c}_{\dot{\mathbf{q}}} + 2\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}$$
$$+ \left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} + \ddot{\boldsymbol{\Phi}}_{\mathbf{q}} - \mathbf{c}_{\mathbf{q}} + \frac{\mathrm{d}\mathbf{c}_{\dot{\mathbf{q}}}}{\mathrm{d}t}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} = g_{\mathbf{q}}^{\mathrm{T}} - \frac{\mathrm{d}g_{\dot{\mathbf{q}}}^{\mathrm{T}}}{\mathrm{d}t} + \frac{\mathrm{d}^2 g_{\ddot{\mathbf{q}}}^{\mathrm{T}}}{\mathrm{d}t^2} \tag{1.51a}$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu} = g_{\boldsymbol{\lambda}} \tag{1.51b}$$

The adjoint system (1.51a), (1.51b) can be easily proved to be an index-1 DAE in $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$. The initial conditions for the adjoint DAE are:

$$\left[\mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \mathbf{c}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right]_F$$
$$= \left[-w_{\mathbf{q}}^{\mathrm{T}} - g_{\dot{\mathbf{q}}}^{\mathrm{T}} + \frac{\mathrm{d}g_{\ddot{\mathbf{q}}}^{\mathrm{T}}}{\mathrm{d}t}\right]_F \tag{1.52a}$$

$$\left[\mathbf{M}^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right]_F = \left[\left(w_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\right)^{\mathrm{T}}\right]_F \tag{1.52b}$$

$$\left[w_{\ddot{\mathbf{q}}}\right]_F = \mathbf{0} \tag{1.52c}$$

$$\left[w_{\boldsymbol{\lambda}}\right]_F = \mathbf{0} \tag{1.52d}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu}\right]_F = \left[g_{\boldsymbol{\lambda}}^{\mathrm{T}}\right]_F \tag{1.52e}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\dot{\boldsymbol{\mu}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\boldsymbol{\mu}\right]_F = \left[\frac{\mathrm{d}g_{\boldsymbol{\lambda}}^{\mathrm{T}}}{\mathrm{d}t}\right]_F \tag{1.52f}$$

Observe that two incompatibility conditions (1.52c) and (1.52d), arise for the objective functionals whose final time term cannot depend on $\ddot{\mathbf{q}}$ or $\boldsymbol{\lambda}$. Moreover, the extra condition (1.52e) and its derivative (1.52f) were taken from (1.51b) particularized for the final time to complete a full set of equations, which allow to obtain the initial values of $\boldsymbol{\mu}$, $\dot{\boldsymbol{\mu}}$, $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ and $\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}$ at the final time t_F.

The following gradient of the objective function is obtained from the variational equation (1.50):

$$\nabla_{\boldsymbol{\rho}}\psi = \left[w_{\boldsymbol{\rho}}^{\mathrm{T}}\Big|_F - \left[\frac{\partial \mathbf{q}}{\partial \boldsymbol{\rho}}^{\mathrm{T}}\left(g_{\dot{\mathbf{q}}}^{\mathrm{T}} - \frac{\mathrm{d}g_{\ddot{\mathbf{q}}}^{\mathrm{T}}}{\mathrm{d}t} + \left(\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\mathbf{c}_{\dot{\mathbf{q}}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}\right)\right]_0\right.$$
$$- \left[\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\rho}}^{\mathrm{T}}\left(g_{\ddot{\mathbf{q}}}^{\mathrm{T}} - \mathbf{M}^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right)\right]_0 + \int_{t_0}^{t_F}\left(g_{\boldsymbol{\rho}}^{\mathrm{T}} - \left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right)^{\mathrm{T}}\boldsymbol{\mu}\right.$$
$$\left.\left. - \left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right)\mathrm{d}t \right. \tag{1.53}$$

The gradient depends on the solution of the EOM and on the adjoint differential and algebraic variables $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\Phi}}$ which are the solution of the adjoint system.

1.4.2.2 Approach 2

Equations (1.51a), (1.51b) and (1.53), are of theoretical interest, but of lower practical value because they can involve higher order derivatives of the state variables, as pointed out in [20]. A more convenient approach is to use Eqs. (1.8a), (1.8b) to express the variations $\delta\ddot{\mathbf{q}}$ and $\delta\boldsymbol{\lambda}$ in terms of the variations of the states $\delta\mathbf{q}$ and $\delta\dot{\mathbf{q}}$. Taking the variation of (1.8a), (1.8b)

$$\begin{bmatrix}\delta\ddot{\mathbf{q}}\\ \delta\boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}\end{bmatrix}^{-1}\left(\begin{bmatrix}-\mathbf{K} - \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}\\ \mathbf{c}_{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}}\end{bmatrix}\delta\mathbf{q} + \begin{bmatrix}-\mathbf{C}\\ \mathbf{c}_{\dot{\mathbf{q}}}\end{bmatrix}\delta\dot{\mathbf{q}}\right.$$
$$\left. + \begin{bmatrix}\mathbf{Q}_{\boldsymbol{\rho}} - \mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda}\\ \mathbf{c}_{\boldsymbol{\rho}} - \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}}\end{bmatrix}\delta\boldsymbol{\rho}\right) \tag{1.54}$$

$$\delta\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\mathbf{q}}\delta\mathbf{q} + \ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + \ddot{\mathbf{q}}_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}, \tag{1.55a}$$
$$\delta\boldsymbol{\lambda} = \boldsymbol{\lambda}_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\lambda}_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + \boldsymbol{\lambda}_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}, \tag{1.55b}$$

where

$$\begin{bmatrix}\ddot{\mathbf{q}}_{\mathbf{q}}\\ \boldsymbol{\lambda}_{\mathbf{q}}\end{bmatrix} = \begin{bmatrix}\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}-\mathbf{K} - \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}\\ \mathbf{c}_{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}}\end{bmatrix}, \tag{1.56}$$

$$\begin{bmatrix}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\\ \boldsymbol{\lambda}_{\dot{\mathbf{q}}}\end{bmatrix} = \begin{bmatrix}\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}-\mathbf{C}\\ \mathbf{c}_{\dot{\mathbf{q}}}\end{bmatrix}, \tag{1.57}$$

$$\begin{bmatrix}\ddot{\mathbf{q}}_{\boldsymbol{\rho}}\\ \boldsymbol{\lambda}_{\boldsymbol{\rho}}\end{bmatrix} = \begin{bmatrix}\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{Q}_{\boldsymbol{\rho}} - \mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda}\\ \mathbf{c}_{\boldsymbol{\rho}} - \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}}\end{bmatrix}. \tag{1.58}$$

Replacing Eqs. (1.55a) and (1.55b) in (1.49) yields

$$
\begin{aligned}
\delta\mathcal{L} = & \left[\left(w_{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}}\right)\delta\mathbf{q} + \left(w_{\dot{\mathbf{q}}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)\delta\dot{\mathbf{q}} + \left(w_{\rho} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\rho}\right)\delta\boldsymbol{\rho}\right]_{t_F} \\
& + \int_{t_0}^{t_F}\left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} - \mathbf{c}_{\mathbf{q}}\right)\right)\delta\mathbf{q}\mathrm{d}t \\
& + \int_{t_0}^{t_F}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{C} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}}\right)\delta\dot{\mathbf{q}}\mathrm{d}t - \int_{t_0}^{t_F}\left(\boldsymbol{\mu}^{\mathrm{T}}\mathbf{M} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}\right)\delta\ddot{\mathbf{q}}\mathrm{d}t - \int_{t_0}^{t_F}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\delta\boldsymbol{\lambda}\mathrm{d}t \\
& + \int_{t_0}^{t_F}\left(g_{\boldsymbol{\rho}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)\right)\delta\boldsymbol{\rho}\mathrm{d}t
\end{aligned}
\tag{1.59}
$$

Observe that only the variations $\delta\ddot{\mathbf{q}}$ and $\delta\boldsymbol{\lambda}$ associated with the objective function terms w and g were eliminated in (1.59). Integration by parts can be applied to the remaining $\delta\ddot{\mathbf{q}}$ and $\delta\dot{\mathbf{q}}$ terms

$$
\begin{aligned}
\delta\mathcal{L} = & \left[\left(w_{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}}\right)\delta\mathbf{q} + \left(w_{\dot{\mathbf{q}}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)\delta\dot{\mathbf{q}} + \left(w_{\rho} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\rho}\right)\delta\boldsymbol{\rho}\right]_{t_F} \\
& \int_{t_0}^{t_F}\left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K}\right) - \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} - \mathbf{c}_{\mathbf{q}}\right)\right)\delta\mathbf{q}\mathrm{d}t \\
& + \left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\mathbf{C} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} \\
& - \int_{t_0}^{t_F}\left(\frac{\mathrm{d}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)}{\mathrm{d}t} - \dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{C} - \boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{C}} + \dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\mathbf{c}_{\dot{\mathbf{q}}} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{c}_{\dot{\mathbf{q}}}}{\mathrm{d}t}\right)\delta\mathbf{q}\mathrm{d}t \\
& - \left(\boldsymbol{\mu}^{\mathrm{T}}\mathbf{M} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}\right)\delta\dot{\mathbf{q}}\Big|_{t0}^{tF} + \left(\dot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M} + \boldsymbol{\mu}^{\mathrm{T}}\dot{\mathbf{M}} + \dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} \\
& - \int_{t_0}^{t_F}\left(\ddot{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{M} + 2\dot{\boldsymbol{\mu}}^{\mathrm{T}}\dot{\mathbf{M}} + \boldsymbol{\mu}^{\mathrm{T}}\ddot{\mathbf{M}} + \ddot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}} + 2\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}^{\mathrm{T}}\dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\ddot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\delta\mathbf{q}\mathrm{d}t \\
& - \int_{t_0}^{t_F}\left(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\right)\delta\boldsymbol{\lambda}\mathrm{d}t + \int_{t_0}^{t_F}\left(g_{\boldsymbol{\rho}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right)\right. \\
& \left.-\boldsymbol{\mu}_{\boldsymbol{\Phi}}^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)\right)\delta\boldsymbol{\rho}\mathrm{d}t
\end{aligned}
\tag{1.60}
$$

Canceling all the integral terms that involve $\delta\mathbf{q}$ and $\delta\boldsymbol{\lambda}$ leads to the following adjoint DAE

$$
\begin{aligned}
& \mathbf{M}^{\mathrm{T}}\ddot{\boldsymbol{\mu}} + \left(2\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda} + \mathbf{K} - \dot{\mathbf{C}} + \ddot{\mathbf{M}}\right)^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\ddot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}} + \left(\mathbf{c}_{\dot{\mathbf{q}}} + 2\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}} \\
& + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} + \ddot{\boldsymbol{\Phi}}_{\mathbf{q}} - \mathbf{c}_{\mathbf{q}} + \frac{\mathrm{d}\mathbf{c}_{\dot{\mathbf{q}}}}{\mathrm{d}t}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} = \left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}}\right)^{\mathrm{T}} - \frac{\mathrm{d}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}}}{\mathrm{d}t}
\end{aligned}
\tag{1.61a}
$$

$$\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu} = \mathbf{0}. \tag{1.61b}$$

Observe that the last term in equation (1.61a) can be difficult to obtain, because the temporal derivatives of a functional which depends on the accelerations and Lagrange multipliers can involve temporal derivatives of the accelerations and temporal derivatives of the Lagrange multipliers which normally are not calculated by the integrator.

The initial conditions for the adjoint are the following

$$\begin{aligned}\left[\mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \mathbf{c}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right]_F \\ = -\left[\left(w_{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\mathbf{q}} + g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}}\right]_F\end{aligned} \tag{1.62a}$$

$$\left[\mathbf{M}^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right]_F = \left[\left(w_{\dot{\mathbf{q}}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}}\right]_F \tag{1.62b}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu}\right]_F = \mathbf{0} \tag{1.62c}$$

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}\dot{\boldsymbol{\mu}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\boldsymbol{\mu}\right]_F = \mathbf{0} \tag{1.62d}$$

Finally, the gradient

$$\begin{aligned}\nabla_{\boldsymbol{\rho}}\psi = &\left[\left(w_{\boldsymbol{\rho}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + w_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}}\right)^{\mathrm{T}}\right]_F \\ &- \left[\frac{\partial \mathbf{q}}{\partial \boldsymbol{\rho}}^{\mathrm{T}}\left(\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}\right)^{\mathrm{T}} + \left(\dot{\mathbf{M}} - \mathbf{C}\right)^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{M}^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\mathbf{c}_{\dot{\mathbf{q}}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\dot{\boldsymbol{\mu}}_{\boldsymbol{\Phi}}\right)\right]_0 \\ &+ \left[\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\rho}}^{\mathrm{T}}\left(\mathbf{M}^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right)\right]_0 + \int_{t_0}^{t_F}\left(\left(g_{\boldsymbol{\rho}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\boldsymbol{\rho}} + g_{\boldsymbol{\lambda}}\boldsymbol{\lambda}_{\boldsymbol{\rho}}\right)^{\mathrm{T}} - \left(\mathbf{M}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}^{\mathrm{T}}\boldsymbol{\lambda} - \mathbf{Q}_{\boldsymbol{\rho}}\right)^{\mathrm{T}}\boldsymbol{\mu}\right. \\ &\left. - \left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} - \mathbf{c}_{\boldsymbol{\rho}}\right)^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{\Phi}}\right)\mathrm{dt}\end{aligned} \tag{1.63}$$

1.4.3 Adjoint Sensitivity: Penalty Formulation

Considering the equations of motion (1.10a), the Lagrangian in this case has the following expression

$$\mathcal{L}(\boldsymbol{\rho}) = w\left(\mathbf{q}_F, \dot{\mathbf{q}}_F, \ddot{\mathbf{q}}_F, \boldsymbol{\rho}_F, \boldsymbol{\lambda}_F\right) + \int_{t_0}^{t_F} g\left(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}^*, \boldsymbol{\rho}\right) \mathrm{dt}$$
$$- \int_{t_0}^{t_F} \boldsymbol{\mu}^{\mathrm{T}} \left(\bar{\mathbf{M}}\left(\mathbf{q}, \boldsymbol{\rho}\right) \ddot{\mathbf{q}} - \bar{\mathbf{Q}}\left(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\rho}\right)\right) \mathrm{dt} \tag{1.64}$$

Note that in this case the Lagrange multipliers are not present for the formulation considered here, but they can be approximated by $\boldsymbol{\lambda}^*$ by means of Eq. (1.11), making possible to consider dependencies of the objective function with the constraint forces. Note that since $\boldsymbol{\lambda}^*$ is not a variable in the equations of motion the dependency of the cost function on $\boldsymbol{\lambda}^*$ is formal and used for convenience. The same final sensitivity equations are obtained when $\boldsymbol{\lambda}^*$ is not included in (1.64) and the derivatives with respect to $\boldsymbol{\lambda}^*$ are obtained from post-processing the adjoint variables.

Applying variational calculus

$$\begin{aligned}\delta\mathcal{L} = &\left[w_{\mathbf{q}}\delta\mathbf{q} + w_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}} + w_{\boldsymbol{\lambda}^*}\delta\boldsymbol{\lambda}^* + w_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}\right]_{t_F} \\ &+ \int_{t_0}^{t_F} \left(g_{\mathbf{q}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}} + \bar{\mathbf{K}}\right)\right)\delta\mathbf{q}\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\dot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{C}}\right)\delta\dot{\mathbf{q}}\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\ddot{\mathbf{q}}} - \boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{M}}\right)\delta\ddot{\mathbf{q}}\mathrm{dt} \\ &+ \int_{t_0}^{t_F} g_{\boldsymbol{\lambda}^*}\delta\boldsymbol{\lambda}^*\mathrm{dt} + \int_{t_0}^{t_F} \left(g_{\boldsymbol{\rho}} - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\boldsymbol{\rho}}\ddot{\mathbf{q}} - \bar{\mathbf{Q}}_{\boldsymbol{\rho}}\right)\right)\delta\boldsymbol{\rho}\mathrm{dt}\end{aligned} \tag{1.65}$$

The variation $\delta\boldsymbol{\lambda}^*$ can be removed by expressing it in terms of the variations $\delta\mathbf{q}$, $\delta\dot{\mathbf{q}}$ and $\delta\ddot{\mathbf{q}}$. From Eq. (1.11)

$$\delta\boldsymbol{\lambda}^* = \boldsymbol{\alpha}\left(\delta\ddot{\boldsymbol{\Phi}} + 2\xi\omega\delta\dot{\boldsymbol{\Phi}} + \omega^2\delta\boldsymbol{\Phi}\right) \tag{1.66}$$

where

$$\begin{aligned}\delta\ddot{\boldsymbol{\Phi}} = &\boldsymbol{\Phi}_{\mathbf{q}}\delta\ddot{\mathbf{q}} + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}}\right)\delta\dot{\mathbf{q}} \\ &+ \left(\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t}\right)_{\mathbf{q}}\right)\delta\mathbf{q} \\ &+ \left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)_{\boldsymbol{\rho}}\dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t}\right)_{\boldsymbol{\rho}}\right)\delta\boldsymbol{\rho}\end{aligned} \tag{1.67}$$

$$\delta\dot{\boldsymbol{\Phi}} = \boldsymbol{\Phi}_{\mathbf{q}}\delta\dot{\mathbf{q}} + \left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}}\right)\delta\mathbf{q} + \left(\boldsymbol{\Phi}_{\mathbf{q}\boldsymbol{\rho}}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\boldsymbol{\rho}}\right)\delta\boldsymbol{\rho} \tag{1.68}$$

$$\delta\boldsymbol{\Phi} = \boldsymbol{\Phi}_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\Phi}_{\boldsymbol{\rho}}\delta\boldsymbol{\rho} \tag{1.69}$$

Grouping together the terms associated to $\delta\ddot{\mathbf{q}}$, $\delta\dot{\mathbf{q}}$, $\delta\mathbf{q}$, $\delta\boldsymbol{\rho}$, Eq. (1.66) becomes

$$\delta\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\delta\ddot{\mathbf{q}} + \boldsymbol{\lambda}^*_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + \boldsymbol{\lambda}^*_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\lambda}^*_{\boldsymbol{\rho}}\delta\boldsymbol{\rho}, \tag{1.70}$$

Identifying the common terms in (1.66) and (1.70)

$$\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^{*} = \alpha \boldsymbol{\Phi}_{\mathbf{q}} \tag{1.71a}$$

$$\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^{*} = \alpha \left[\boldsymbol{\Phi}_{\mathbf{qq}} \dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}} + 2\xi\omega \boldsymbol{\Phi}_{\mathbf{q}} \right] \tag{1.71b}$$

$$\begin{aligned} \boldsymbol{\lambda}_{\mathbf{q}}^{*} = \alpha \Big[& \boldsymbol{\Phi}_{\mathbf{qq}} \ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}} \right)_{\mathbf{q}} \dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t} \right)_{\mathbf{q}} \\ & + 2\xi\omega \left(\boldsymbol{\Phi}_{\mathbf{qq}} \dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\mathbf{q}} \right) + \omega^{2} \boldsymbol{\Phi}_{\mathbf{q}} \Big] \end{aligned} \tag{1.71c}$$

$$\begin{aligned} \boldsymbol{\lambda}_{\rho}^{*} = \alpha \Big[& \boldsymbol{\Phi}_{\mathbf{q}\rho} \ddot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}} \right)_{\rho} \dot{\mathbf{q}} + \left(\dot{\boldsymbol{\Phi}}_{t} \right)_{\rho} \\ & + 2\xi\omega \left(\boldsymbol{\Phi}_{\mathbf{q}\rho} \dot{\mathbf{q}} + \boldsymbol{\Phi}_{t\rho} \right) + \omega^{2} \boldsymbol{\Phi}_{\rho} \Big] \end{aligned} \tag{1.71d}$$

The variation $\delta\ddot{\mathbf{q}}$ can be removed too, by expressing it in terms of the variations $\delta\mathbf{q}$, $\delta\dot{\mathbf{q}}$ and $\delta\ddot{\mathbf{q}}$. Making use of Eqs. (1.9) and (1.70)

$$\mathbf{M}\delta\ddot{\mathbf{q}} + \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\delta\boldsymbol{\lambda}^{*} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}^{*}\delta\mathbf{q} + \boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}^{*}\delta\boldsymbol{\rho} = \mathbf{Q}_{\mathbf{q}}\delta\mathbf{q} + \mathbf{Q}_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + \mathbf{Q}_{\rho}\delta\boldsymbol{\rho} \tag{1.72}$$

$$\begin{aligned} \bar{\mathbf{M}}\delta\ddot{\mathbf{q}} = & -\left(\mathbf{K} + \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathbf{q}}^{*} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} \right)\delta\mathbf{q} - \left(\mathbf{C} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^{*} \right)\delta\dot{\mathbf{q}} \\ & + \left(\mathbf{Q}_{\rho} - \mathbf{M}_{\rho}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\rho}^{*} - \boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}^{*} \right)\delta\boldsymbol{\rho} \end{aligned} \tag{1.73}$$

Then

$$\delta\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\mathbf{q}}\delta\mathbf{q} + \ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} + \ddot{\mathbf{q}}_{\rho}\delta\boldsymbol{\rho} \tag{1.74}$$

where

$$\ddot{\mathbf{q}}_{\mathbf{q}} = -\bar{\mathbf{M}}^{-1}\left(\mathbf{K} + \mathbf{M}_{\mathbf{q}}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\mathbf{q}}^{*} + \boldsymbol{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\boldsymbol{\lambda}^{*} \right) \tag{1.75}$$

$$\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} = -\bar{\mathbf{M}}^{-1}\left(\mathbf{C} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^{*} \right) \tag{1.76}$$

$$\ddot{\mathbf{q}}_{\rho} = \bar{\mathbf{M}}^{-1}\left(\mathbf{Q}_{\rho} - \mathbf{M}_{\rho}\ddot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}_{\rho}^{*} - \boldsymbol{\Phi}_{\mathbf{q}\rho}^{\mathrm{T}}\boldsymbol{\lambda}^{*} \right) \tag{1.77}$$

Replacing Eq. (1.74) back in Eq. (1.70) yields

$$\delta\boldsymbol{\lambda}^{*} = \left(\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^{*} + \boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^{*}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} \right)\delta\dot{\mathbf{q}} + \left(\boldsymbol{\lambda}_{\mathbf{q}}^{*} + \boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^{*}\ddot{\mathbf{q}}_{\mathbf{q}} \right)\delta\mathbf{q} + \left(\boldsymbol{\lambda}_{\rho}^{*} + \boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^{*}\ddot{\mathbf{q}}_{\rho} \right)\delta\boldsymbol{\rho}, \tag{1.78}$$

After replacing Eqs. (1.74) and (1.78) in (1.65), the variations $\delta\ddot{\mathbf{q}}$ and $\delta\boldsymbol{\lambda}^{*}$ disappear

$$\begin{aligned}\delta\mathcal{L} = &\Big[\left(w_{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\mathbf{q}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}\right)\right)\delta\mathbf{q} + \left(w_{\dot{\mathbf{q}}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\right)\delta\dot{\mathbf{q}} \\ &+ \left(w_{\rho} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\rho} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho}\right)\right)\delta\boldsymbol{\rho}\Big]_F \\ &+ \int_{t_0}^{t_F}\left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\mathbf{q}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}\right) - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}} + \bar{\mathbf{K}}\right)\right)\delta\mathbf{q}\mathrm{d}t \\ &+ \int_{t_0}^{t_F}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right) - \boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{C}}\right)\delta\dot{\mathbf{q}}\mathrm{d}t \\ &- \int_{t_0}^{t_F}\boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{M}}\delta\ddot{\mathbf{q}}\mathrm{d}t + \int_{t_0}^{t_F}\left(g_{\rho} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\rho} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho}\right) - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\rho}\ddot{\mathbf{q}} - \bar{\mathbf{Q}}_{\rho}\right)\right)\delta\boldsymbol{\rho}\mathrm{d}t\end{aligned} \tag{1.79}$$

Integrating by parts the integrals involving $\delta\dot{\mathbf{q}}$, $\delta\ddot{\mathbf{q}}$:

$$\begin{aligned}\delta\mathcal{L} = &\Big[\left(w_{\mathbf{q}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\mathbf{q}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}\right)\right)\delta\mathbf{q} + \left(w_{\dot{\mathbf{q}}} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\right)\delta\dot{\mathbf{q}} \\ &+ \left(w_{\rho} + w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + w_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\rho} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho}\right)\right)\delta\boldsymbol{\rho}\Big]_F \\ &+ \int_{t_0}^{t_F}\left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\mathbf{q}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}\right) - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}} + \bar{\mathbf{K}}\right)\right)\delta\mathbf{q}\mathrm{d}t \\ &+ \left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right) - \boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{C}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} \\ &- \int_{t_0}^{t_F}\left(\frac{\mathrm{d}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\right)}{\mathrm{d}t} - \dot{\boldsymbol{\mu}}^{\mathrm{T}}\bar{\mathbf{C}} - \boldsymbol{\mu}^{\mathrm{T}}\dot{\bar{\mathbf{C}}}\right)\delta\mathbf{q}\mathrm{d}t \\ &- \left(\boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{M}}\right)\delta\dot{\mathbf{q}}\Big|_{t0}^{tF} + \left(\dot{\boldsymbol{\mu}}^{\mathrm{T}}\bar{\mathbf{M}} + \boldsymbol{\mu}^{\mathrm{T}}\dot{\bar{\mathbf{M}}}\right)\delta\mathbf{q}\Big|_{t0}^{tF} - \int_{t_0}^{t_F}\left(\ddot{\boldsymbol{\mu}}^{\mathrm{T}}\bar{\mathbf{M}} + 2\dot{\boldsymbol{\mu}}^{\mathrm{T}}\dot{\bar{\mathbf{M}}} + \boldsymbol{\mu}^{\mathrm{T}}\ddot{\bar{\mathbf{M}}}\right)\delta\mathbf{q}\mathrm{d}t \\ &+ \int_{t_0}^{t_F}\left(g_{\rho} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\rho} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\rho}\right) - \boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_{\rho}\ddot{\mathbf{q}} - \bar{\mathbf{Q}}_{\rho}\right)\right)\delta\boldsymbol{\rho}\mathrm{d}t\end{aligned} \tag{1.80}$$

Canceling all the integral terms that involve $\delta\mathbf{q}$ leads to the following adjoint ODE

$$\begin{aligned}\bar{\mathbf{M}}^{\mathrm{T}}\ddot{\boldsymbol{\mu}} + \left(2\dot{\bar{\mathbf{M}}} - \bar{\mathbf{C}}\right)^{\mathrm{T}}\dot{\boldsymbol{\mu}} + \left(\bar{\mathbf{M}}_{\mathbf{q}}\ddot{\mathbf{q}} + \bar{\mathbf{K}} - \dot{\bar{\mathbf{C}}} + \ddot{\bar{\mathbf{M}}}\right)^{\mathrm{T}}\boldsymbol{\mu} = &\left(g_{\mathbf{q}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\mathbf{q}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}\right)\right)^{\mathrm{T}} \\ &- \frac{\mathrm{d}\left(g_{\dot{\mathbf{q}}} + g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}} + g_{\lambda^*}\left(\boldsymbol{\lambda}^*_{\dot{\mathbf{q}}} + \boldsymbol{\lambda}^*_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\right)^{\mathrm{T}}}{\mathrm{d}t}\end{aligned} \tag{1.81a}$$

$$\left[\bar{\mathbf{M}}^{\mathrm{T}}\dot{\boldsymbol{\mu}}+\left(\dot{\bar{\mathbf{M}}}-\dot{\bar{\mathbf{C}}}\right)^{\mathrm{T}}\boldsymbol{\mu}\right]_F=-\Big[w_{\mathbf{q}}+w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\mathbf{q}}+w_{\lambda^*}\left(\boldsymbol{\lambda}_{\mathbf{q}}^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_{\mathbf{q}}\right)+g_{\dot{\mathbf{q}}}+g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}$$
$$+g_{\lambda^*}\left(\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\Big]_F^{\mathrm{T}} \tag{1.81b}$$

$$\left[\bar{\mathbf{M}}^{\mathrm{T}}\boldsymbol{\mu}\right]_F=\left[w_{\dot{\mathbf{q}}}+w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}+w_{\lambda^*}\left(\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)\right]_F^{\mathrm{T}} \tag{1.81c}$$

Finally, the gradient can be obtained from the remaining terms in the variational equation (1.80)

$$\begin{aligned}\nabla_\rho\psi=&\left[w_\rho+w_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_\rho+w_{\lambda^*}\left(\boldsymbol{\lambda}_\rho^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_\rho\right)\right]_F\\&-\left[\mathbf{q}_\rho^{\mathrm{T}}\left(g_{\dot{\mathbf{q}}}+g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}+g_{\lambda^*}\left(\boldsymbol{\lambda}_{\dot{\mathbf{q}}}^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_{\dot{\mathbf{q}}}\right)+\dot{\boldsymbol{\mu}}^{\mathrm{T}}\bar{\mathbf{M}}+\boldsymbol{\mu}^{\mathrm{T}}\left(\dot{\bar{\mathbf{M}}}-\dot{\bar{\mathbf{C}}}\right)\right)^{\mathrm{T}}\right]_{t0}\\&+\left[\dot{\mathbf{q}}_\rho^{\mathrm{T}}\left(\boldsymbol{\mu}^{\mathrm{T}}\bar{\mathbf{M}}\right)^{\mathrm{T}}\right]_{t0}+\int_{t_0}^{t_F}\left(g_\rho+g_{\ddot{\mathbf{q}}}\ddot{\mathbf{q}}_\rho+g_{\lambda^*}\left(\boldsymbol{\lambda}_\rho^*+\boldsymbol{\lambda}_{\ddot{\mathbf{q}}}^*\ddot{\mathbf{q}}_\rho\right)-\boldsymbol{\mu}^{\mathrm{T}}\left(\bar{\mathbf{M}}_\rho\ddot{\mathbf{q}}-\bar{\mathbf{Q}}_\rho\right)\right)^{\mathrm{T}}\mathrm{d}t\end{aligned} \tag{1.82}$$

Observe that more derivatives than those already obtained for the direct sensitivity approach are necessary in (1.81a), (1.81b), (1.81c) and (1.82). The additional terms not obtained before are $\dot{\bar{\mathbf{M}}}$, $\ddot{\bar{\mathbf{M}}}$ and $\dot{\bar{\mathbf{C}}}$.

From equation (1.10b)

$$\dot{\bar{\mathbf{M}}}=\dot{\mathbf{M}}+\dot{\boldsymbol{\Phi}}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}}, \tag{1.83}$$

$$\ddot{\bar{\mathbf{M}}}=\ddot{\mathbf{M}}+\ddot{\boldsymbol{\Phi}}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}+2\dot{\boldsymbol{\Phi}}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\ddot{\boldsymbol{\Phi}}_{\mathbf{q}} \tag{1.84}$$

and from Eq. (1.30b)

$$\begin{aligned}\dot{\bar{\mathbf{C}}}=&\dot{\mathbf{C}}+\dot{\boldsymbol{\Phi}}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\boldsymbol{\Phi}_{\mathbf{qq}}\dot{\mathbf{q}}+\dot{\boldsymbol{\Phi}}_{\mathbf{q}}+\boldsymbol{\Phi}_{t\mathbf{q}}+2\xi\omega\boldsymbol{\Phi}_{\mathbf{q}}\right)\\&+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{qq}}\dot{\mathbf{q}}+\boldsymbol{\Phi}_{\mathbf{qq}}\ddot{\mathbf{q}}+\ddot{\boldsymbol{\Phi}}_{\mathbf{q}}+\dot{\boldsymbol{\Phi}}_{t\mathbf{q}}+2\xi\omega\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\end{aligned} \tag{1.85}$$

1.4.4 Validation of the Computed Sensitivities

The validation of the computed sensitivities is crucial because small errors in individual terms can result in completely wrong sensitivities and even if wrong sensitivities can usually solve optimization problems, they are still wrong. The strategies proposed and employed here to validate sensitivities are the following:

1. Compare the results of direct and adjoint sensitivity approaches: they should be equal within the truncation error.
2. Compare the results of different formulations of the equations of motion: Index-3, Index-1 and Penalty sensitivities were compared.

3. Compute the sensitivities using a third party code: FATODE [23] was used to double-check the results presented.
4. Use *real* finite differences to approximate whole sensitivities or individual derivatives: it can be a very inaccurate or even completely useless strategy. Small truncation errors (small δ) cause important loss-of-significance errors. The first-order approximation of the derivatives with real perturbations reads

$$\frac{\mathrm{d}\psi}{\mathrm{d}\rho_k} = \frac{\psi\left(\boldsymbol{\rho} + \delta \mathbf{e}_k\right) - \psi\left(\boldsymbol{\rho}\right)}{\delta}. \tag{1.86}$$

   The truncation error in this case is $\mathcal{O}(\delta)$ cf. (1.86) and the loss of significance errors are order $\mathcal{O}(\delta^{-1})$, where δ is the perturbation. This fact can make these derivatives highly inaccurate.
5. Use *complex* finite differences to approximate whole sensitivities or individual derivatives: it is a much more reliable approach than the previous one, but more complex to implement. Since there is not subtraction, there are not loss-of-significance errors in the imaginary part. The first-order approximation of derivatives with complex perturbations is the following

$$\frac{\mathrm{d}\psi}{\mathrm{d}\rho_k} = \frac{\Im\left(\psi\left(\boldsymbol{\rho} + i\,\delta \mathbf{e}_k\right)\right)}{\delta}, \tag{1.87}$$

   where i is the imaginary unit and $\Im$ is the imaginary part of a complex number. The approach is considerably more accurate than the previous one, because there are no subtractions in the imaginary parts and therefore the perturbations can be chosen arbitrarily small without loss-of-significance errors appearing in the calculation of the approximation. The practical difficulty to apply complex finite differences is that not all codes can be changed easily to accommodate complex arithmetic. Special attention should be paid to the third party functions (if any) involved in the code (*transpose* functions, *norm* functions, numerical integrator chosen, etc).

1.5 Numerical Experiment

The mechanism chosen to test the formulations proposed in the paper is the five bar mechanism with 2 degrees of freedom shown in Fig. 1.1. The five bars are constrained by five revolute joints located in points A, 1, 2, 3 and B. The five bars are constrained by five revolute joints located in points A, 1, 2, 3 and B. The masses of the bars are $m_1 = 1\,\mathrm{kg}$, $m_2 = 1.5\,\mathrm{kg}$, $m_3 = 1.5\,\mathrm{kg}$, $m_4 = 1\,\mathrm{kg}$ and the polar moments of inertia are calculated under the assumption of a uniform distribution of mass. The mechanism is subjected to the action of gravity and two elastic forces coming from the springs. The stiffness coefficients of the springs are $k_1 = k_2 = 100\,\mathrm{N/m}$ and their

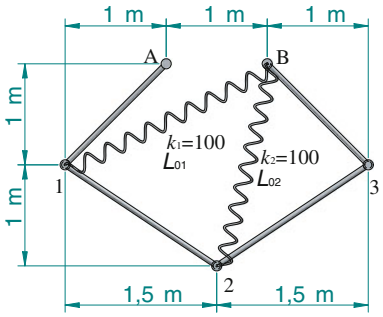


Fig. 1.1 The five-bar mechanism

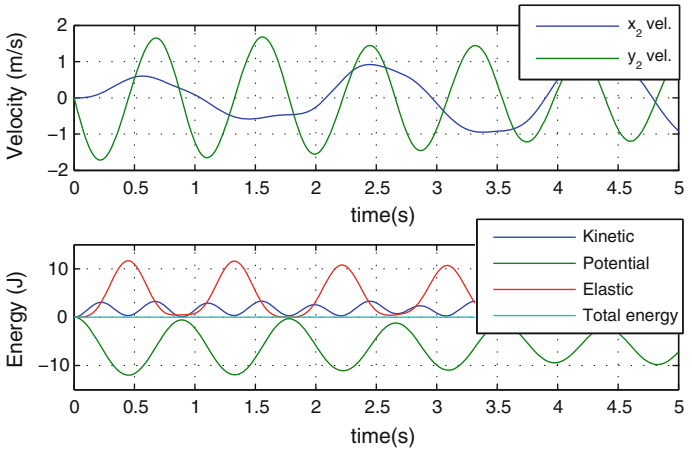


Fig. 1.2 Mechanism response: *Top* velocity of point 2; *Bottom* energy of the system

natural lengths are initially chosen $L_{01} = \sqrt{2^2 + 1^2}$ m and $L_{02} = \sqrt{2^2 + 0.5^2}$ m, coincident with the initial configuration shown in Fig. 1.1.

The response of the system is shown in Fig. 1.2 for a 5 s simulation. The upper plot represents the horizontal and vertical velocities of the point 2 while the lower one represents the energy taking as reference for the potential energy the initial configuration of the system.

The following objective function was chosen:

$$\psi = \int_{t_0}^{t_F} (\mathbf{r}_2 - \mathbf{r}_{20})^{\mathrm{T}} (\mathbf{r}_2 - \mathbf{r}_{20}) \, \mathrm{dt} \tag{1.88}$$

Table 1.1 Results for the five bar mechanism

Approach	Parameters	$\mathrm{d}\psi/\mathrm{d}L_{01}$	$\mathrm{d}\psi/\mathrm{d}L_{02}$
1: Direct index-3	$h = 10^{-2}$ s	−4.2381	3.2170
2: Adjoint index-3	$h = 10^{-2}$ s	−4.2287	3.2090
3: Direct index-1	$h = 10^{-2}$ s	−4.2383	3.2169
4: Adjoint index-1	$h = 10^{-2}$ s	−4.2294	3.2093
5: Direct penalty	$h = 10^{-2}$ s	−4.2305	3.2154
6: Adjoint penalty	$h = 10^{-2}$ s	−4.2299	3.2134
7: FATODE	$Tol = 10^{-3}$	−4.2257	3.2077
8: Num. diff. real with penalty	$\delta = 10^{-7}$ m	−9.7390	−4.0344
9: Num. diff. complex with penalty	$\delta/i = 10^{-7}$ m	−4.2288	3.2116

where $\mathbf{r}_2$ is the global position of the point 2 and $\mathbf{r}_{20}$ is the initial position (at $t = 0$) of the same point. As parameters to obtain the sensitivities, the natural lengths of the springs were chosen $\boldsymbol{\rho}^{\mathrm{T}} = [L_{01}, L_{02}]$.

The gradient $\nabla_{\boldsymbol{\rho}}\psi = \mathbf{0}$ was obtained by the direct and adjoint approaches with all the formulations of the equations of motion proposed in the chapter. Moreover a third party code and the numerical sensitivities were obtained to check the correctness of the results. In sum, the following experiments were carried out:

1. Direct sensitivity with all the formulations proposed.
2. Adjoint sensitivity with all the formulations proposed.
3. Adjoint sensitivity with FATODE .
4. Numerical sensitivity with real perturbations.
5. Numerical sensitivity with complex perturbations.

The results for the sensitivities with the mentioned methods are presented in Table 1.1. All the approaches, except the numerical sensitivities with real perturbations, offer similar results. The numerical sensitivities with real perturbations are not reliable if accurate results for the sensitivities are important for the application to tackle. Given the simplicity of the system proposed, definitive conclusions in terms of efficiency cannot be stated.

The methods presented to compute sensitivities can be employed for the optimization of the mechanism. The mechanism can be balanced by properly selecting the two parameters $\boldsymbol{\rho}^{\mathrm{T}} = [L_{01}, L_{02}]$ in order to keep it in equilibrium. Of course the problem can be solved by means of the static equations but the aim here is doing so by dynamical optimization: the objective is to keep the mechanism still in the initial position, which can be represented mathematically by the minimization of the objective function (1.88). The condition to obtain the minimum is the following:

$$\nabla_{\boldsymbol{\rho}}\psi = \mathbf{0} \tag{1.89}$$

All the methods perform similar to solve the optimization problem. In this case the simulation time was reduced to 1 s and the results for the objective function,

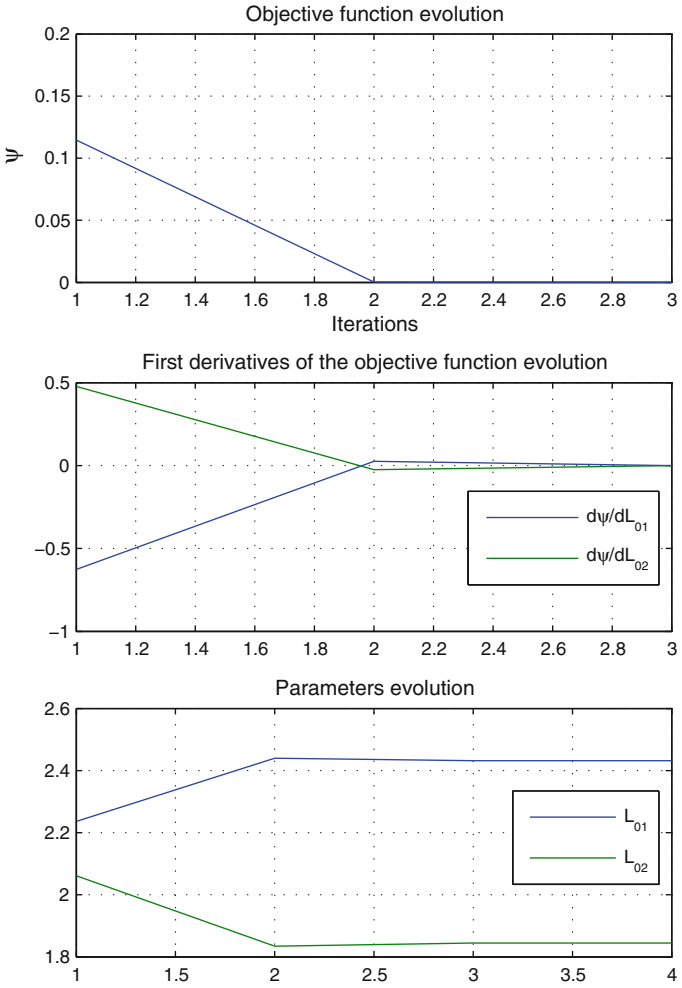


Fig. 1.3 Objective function, gradient and parameters evolution

derivatives and parameters are presented in Fig. 1.3 for the adjoint penalty approach. The plots for the rest of the approaches coincide with the ones presented and they are not presented for clarity.

The optimization converges in three iterations, but in one is almost done. It is important to remark that approximate derivatives can be used to calculate the gradient and the optimization would converge at a lower pace.

Another important remark is that the tolerances in the solution of the forward dynamics are very important in order to obtain stable solutions for the TLM and adjoint ODEs, both of them strongly depend on the solution of the dynamics.

1.6 Conclusions

In this study, the direct and adjoint sensitivities were derived for multibody dynamic systems modeled using several dependent coordinates formulations: index-3 differential-algebraic formulation, index-1differential algebraic formulation and penalty formulation. The objective functions considered depend on the parameters considered, positions, velocities, accelerations and Lagrange multipliers of the system, with the final time fixed. The sensitivity analyses discussed in the context of this work evaluated the gradient of the objective function with respect to the system parameters. It was noticed that the adjoint formulations of the sensitivity equations are not very convenient for the case of objective functions depending on the vector of generalized coordinates or on the Lagrange multipliers, due to the difficulty to obtain the derivatives of these terms, involving higher order derivatives of the states or Lagrange multipliers, normally not calculated.

The validation of the methods developed for sensitivity analysis was conducted on multiple levels: (a) the sensitivities obtained using direct differentiation and adjoint methods were compared; (b) the results of the sensitivities obtained for the different formulations of the equations of motion used in the study were compared; (c) the sensitivities obtained in this study were compared with the results obtained from a third party code, namely FATODE; (d) real finite differences were used to approximate whole sensitivities and individual derivatives (It was observed that this method can be a very inaccurate or even completely useless strategy); (e) complex finite differences were used to approximate whole sensitivities and individual derivative (The approach is considerably more accurate than the previous one. The practical difficulty to apply complex finite differences is that not all codes can be changed easily to accommodate complex arithmetic).

A numerical experiment was conducted on a five-bar mechanism with two degrees of freedom. The following approaches were used on this case study: (a) direct differentiation; (b) adjoint sensitivity; (c) adjoint sensitivity with FATODE; (d) numerical sensitivity with real perturbations; (e) numerical sensitivity with complex perturbations. All the approaches, except the numerical sensitivities with real perturbations, offer similar results which suggests that the schemes proposed are correct. These sensitivities were next used to perform an optimization and all methods performed similarly. It was noticed that approximate derivatives can be used to calculate the gradient and the optimization would converge at a lower pace. Another important remark was that the tolerances in the solution of the forward dynamics are very important in order to obtain stable solutions for the tangent linear models and adjoint ordinary differential equations; both of them strongly depend on this solution.

Acknowledgments This work was supported in part by award NSF CMMI-1130667 and by the Computational Science Laboratory at Virginia Tech.

Appendix: Nomenclature and Differentiation Rules

- $(\cdots)_0$: means evaluation at the initial time $(\cdots)\,(t_0)$.
- $(\cdots)_F$: means evaluation at the final time $(\cdots)\,(t_F)$.
- $\mathbf{q} \in \mathbb{R}^n$: vector of coordinates of the system.
- $\boldsymbol{\rho} \in \mathbb{R}^p$: vector of parameters.
- $\delta\,()$: means variation.
- $()_{\mathbf{q}} = \dfrac{\partial\,()}{\partial \mathbf{q}};\ ()_{\boldsymbol{\rho}} = \dfrac{\partial\,()}{\partial \boldsymbol{\rho}}$
- $\dot{()} = \dfrac{\mathrm{d}\,()}{\mathrm{d}t};\ \ddot{()} = \dfrac{\mathrm{d}^2\,()}{\mathrm{d}t^2}\ ()_t = \dfrac{\partial\,()}{\partial t}$
- $\mathbf{M}\,(\mathbf{q}, \boldsymbol{\rho}) \in \mathbb{R}^{n\times n}$: generalized mass matrix of the system.
- $\mathbf{Q}\,(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\rho}) \in \mathbb{R}^n$: vector of generalized forces of the system.
- $\boldsymbol{\Phi}\,(t, \mathbf{q}, \boldsymbol{\rho}) \in \mathbb{R}^m$: vector of constraints that relate the dependent coordinates.
- $\mathbf{A}_{\mathbf{x}} = \left[\dfrac{\partial \mathbf{A}}{\partial x_1} \cdots \dfrac{\partial \mathbf{A}}{\partial x_i} \cdots \dfrac{\partial \mathbf{A}}{\partial x_s}\right] \in \mathbb{R}^{q\times r\times s}$. Third order tensor of derivatives of matrix $\mathbf{A} \in \mathbb{R}^{q\times r}$ w.r.t. vector $\mathbf{x} \in \mathbb{R}^s$.
- $\mathbf{A}_{\mathbf{x}}^{\mathrm{T}} = \left[\dfrac{\partial \mathbf{A}^{\mathrm{T}}}{\partial x_1} \cdots \dfrac{\partial \mathbf{A}^{\mathrm{T}}}{\partial x_i} \cdots \dfrac{\partial \mathbf{A}^{\mathrm{T}}}{\partial x_s}\right] \in \mathbb{R}^{r\times q\times s}$.
- $\mathbf{A}_{\mathbf{x}}\mathbf{b} = \mathbf{A}_{\mathbf{x}} \otimes \mathbf{b} = \left[\dfrac{\partial \mathbf{A}}{\partial x_1}\mathbf{b} \ldots \dfrac{\partial \mathbf{A}}{\partial x_i}\mathbf{b} \ldots \dfrac{\partial \mathbf{A}}{\partial x_s}\mathbf{b}\right] \in \mathbb{R}^{q\times s}$, where $\mathbf{b} \in \mathbb{R}^r$ is a vector.
- $\mathbf{A}_{\mathbf{x}}\mathbf{B} = \mathbf{A}_{\mathbf{x}} \otimes \mathbf{B} = \left[\dfrac{\partial \mathbf{A}}{\partial x_1}\mathbf{B} \ldots \dfrac{\partial \mathbf{A}}{\partial x_i}\mathbf{B} \ldots \dfrac{\partial \mathbf{A}}{\partial x_s}\mathbf{B}\right] \in \mathbb{R}^{q\times t\times s}$, where $\mathbf{B} \in \mathbb{R}^{r\times t}$ is a matrix.
- $\mathbf{C}\mathbf{A}_{\mathbf{x}}\mathbf{B} = \mathbf{C} \otimes \mathbf{A}_{\mathbf{x}}\mathbf{B} = \left[\mathbf{C}\dfrac{\partial \mathbf{A}}{\partial x_1}\mathbf{B} \ldots \mathbf{C}\dfrac{\partial \mathbf{A}}{\partial x_i}\mathbf{B} \ldots \mathbf{C}\dfrac{\partial \mathbf{A}}{\partial x_s}\mathbf{B}\right] \in \mathbb{R}^{r\times t\times s}$, where $\mathbf{C} \in \mathbb{R}^{r\times q}$ is a matrix.

References

1. Ascher UM, Petzold LR (1998) Computer methods for ordinary differential equations and differential-algebraic equations. Soc Ind Appl Math
2. Brenan KE, Campbell SL, Petzold LR (1989) Numerical solution of initial-value problems in differential-algebraic equations. North-Holland, New York
3. Bottasso CL, Dopico D, Trainelli L (2008) On the optimal scaling of index three daes in multibody dynamics. Multibody Sys Dyn 19(1–2):3–20 cited By (since 1996) 7
4. Bauchau OA, Epple A, Bottasso CL (2009) Scaling of constraints and augmented Lagrangian formulations in multibody dynamics simulations. J Comput Nonlinear Dyn 4, 021007. doi:10.1115/1.3079826
5. Bayo E, García de Jalon J, Serna MA (1988) A modified lagrangian formulation for the dynamic analysis of constrained mechanical systems. Comput Methods Appl Mech Eng 71(2):183–195
6. Bestle D, Seybold J (1992) Sensitivity analysis of constrained multibody systems. Arch Appl Mech 62:181–190

7. Cuadrado J, Cardenal J, Morer P, Bayo E (2000) Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments. Multibody Sys Dyn 4(1):55–73
8. Cuadrado J, Gutierrez R, Naya MA, Morer P (2001) A comparison in terms of accuracy and efficiency between a mbs dynamic formulation with stress analysis and a non-linear fea code. Int J Numer Methods Eng 51(9):1033–1052
9. Ding Jie-Yu, Pan Zhen-Kuan, Chen Li-Qun (2007) Second order adjoint sensitivity analysis of multibody systems described by differential-algebraic equations. Multibody Sys Dyn 18:599–617
10. Dopico D, Zhu Y, Sandu A, Sandu C (2014) Direct and adjoint sensitivity analysis of ODE multibody formulations. J Comput Nonlinear Dyn. doi:10.1115/1.4026492
11. Eich-Soellner E, Führer C (1998) Numerical Methods in Multibody Dynamics. B.G.Teubner, Stuttgart
12. Garcia de Jalon J, Bayo E (1994) Kinematic and dynamic simulation of multibody systems: the real-time challenge. Springer, New York
13. Haug EJ, Arora JS (1979) Applied optimal design: mechanical and structura systems. Wiley, New York
14. Haug E (1987) Design sensitivity analysis of dynamic systems. Number 27 in NATO ASI series. Series F, computer and systems sciences. In: Computer aided optimal design: structural and mechanical systems, Springer, Berlin
15. Haug EJ (1989) Computer aided kinematics and dynamics of mechanical systems: basic methods. Allyn and Bacon. Prentice Hall College Div, Boston
16. Hairer E, Lubich C, Wanner G (2006) Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations, vol 31, 2nd edn. Springer, Berlin (Springer Series in Computational Mathematics)
17. Pagalday JM, Avello A (1997) Optimization of multibody dynamics using object oriented programming and a mixed numerical-symbolic penalty formulation. Mech Mach Theory 32(2):161–174
18. Pagalday JM (1994) Optimizacion del comportamiento dinamico de mecanismos. PhD thesis, Escuela Superior de Ingenieros Industriales de S. Sebastian
19. Petzold L, Lötstedt P (1986) Numerical solution of nonlinear differential equations with algebraic constraints ii: practical implications. SIAM J Sci Stat Comput 7(3):720–733
20. Sonneville Valentin, Brüls Olivier (2014) Sensitivity analysis for multibody systems formulated on a lie group. Multibody Sys Dyn 31(1):47–67
21. Schaffer A (2005) Stability of the adjoint differential-algebraic equation of the index-3 multibody system equation of motion. SIAM J Sci Comput 26(4):1432–1448
22. Schaffer A (2006) Stabilized index-1 differential-algebraic formulations for sensitivity analysis of multi-body dynamics. Proc Inst Mech Eng Part K J Multibody Dyn 220(3):141–156
23. Zhang H, Sandu A (2012) Fatode: a library for forward, adjoint, and tangent linear integration of odes. http://people.cs.vt.edu/~asandu/Software/FATODE/index.html

Chapter 2
A Lagrangian–Lagrangian Framework for the Simulation of Rigid and Deformable Bodies in Fluid

Arman Pazouki, Radu Serban and Dan Negrut

Abstract We present a Lagrangian–Lagrangian approach for the simulation of fully resolved Fluid Solid/Structure Interaction (FSI) problems. In the proposed approach, the method of Smoothed Particle Hydrodynamics (SPH) is used to simulate the fluid dynamics in a Lagrangian framework. The solid phase is a general multibody dynamics system composed of a collection of interacting rigid and deformable objects. While the motion of arbitrarily shaped rigid objects is approached in a classical 3D rigid body dynamics framework, the Absolute Nodal Coordinate Formulation (ANCF) is used to model the deformable components, thus enabling the investigation of compliant elements that experience large deformations with entangling and self-contact. The dynamics of the two phases, fluid and solid, are coupled with the help of Lagrangian markers, referred to as Boundary Condition Enforcing (BCE) markers which are used to impose no-slip and impenetrability conditions. Such BCE markers are associated both with the solid suspended particles and with any confining boundary walls and are distributed in a narrow layer on and below the surface of solid objects. The ensuing fluid–solid interaction forces are mapped into generalized forces on the rigid and flexible bodies and subsequently used to update the dynamics of the solid objects according to rigid body motion or ANCF method. The robustness and performance of the simulation algorithm is demonstrated through several numerical simulation studies.

A. Pazouki (✉) · R. Serban · D. Negrut
University of Wisconsin-Madison, 2035 Mechanical Engineering Building,
1513 University Avenue, Madison, WI 53706, USA
e-mail: pazouki@wisc.edu

R. Serban
e-mail: serban@wisc.edu

D. Negrut
e-mail: negrut@wisc.edu

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_2,

2.1 Introduction

Engineers commonly rely on prototypes and physical testing when performing design and analysis tasks. Unfortunately, such work can be expensive and time consuming. Because computational hardware continues to advance in terms of both processing speed and memory size, a trend is growing in which computer simulation is used to augment and, in some cases, replace large amounts of experimental work. With increasing computational power, engineers are able to perform faster, larger, and more accurate simulations. Computer simulation has several advantages over physical experiments. Through simulation, engineers may study a range of parameter values that would prove too costly or too dangerous to study experimentally. Moreover, computer simulation can produce representative data that experimental measurements could never achieve. Experimental insights are limited by the position, fidelity, and number of sensors, whereas a simulation inherently tracks the state of every component of the system. For example, simulation can generate, in a non-intrusive fashion, the set of forces acting between all the individual bodies in a flow of suspension.

Current simulation capabilities are sometimes inadequate to capture phenomena of interest. This problem is especially evident when simulating the dynamics of Fluid–Solid Interaction (FSI) systems, which may contain tens of thousands of rigid and deformable bodies that interact directly or through the fluid media. The ability to solve such large problems will require significant improvements in terms of both algorithms and implementation.

To alleviate computational limitations, numerical simulation approaches devised for the general category of FSI problems usually suppress some physics depending on the specific application. For instance, several approaches have been proposed to study characteristics of the flow of particle suspension. These include Eulerian–Eulerian (EE) approaches, where the solid phase is considered as a continuum [14, 16, 46]; Lagrangian particle tracking, also known as Lagrangian Numerical Simulation (LNS) approaches, which either consider a one-way coupling of fluid and solid phase, or else introduce a collective momentum exchange term to the fluid equation [2, 30]; Eulerian–Lagrangian (EL) approaches, where the Lagrangian solid phase moves with/within the Eulerian grid used for fluid simulation [17, 21, 25]; and Lagrangian–Lagrangian (LL) approaches, where both phases are modeled within a Lagrangian framework [36, 38, 39]. As in EE methodologies, LNS approaches rely on empirical forms of hydrodynamic fluid–solid forces, determined mostly for dilute conditions where the particle-particle interaction is neglected.

Similar approaches are also applied to the fluid-structure interaction. In this document, the focus is primarily on the LL approaches, particularly those geared towards large deformation favored by the multibody dynamics community (some studies on problems involving small structural deformation using a Lagrangian representation of fluid flow are provided in [1, 4, 28]).

The body of work on FSI problems using Lagrangian fluid representation and large structural deformation is very limited. Schörgenhumer et al. [42] presented

a co-simulation approach for the FSI problems. In their approach, they used a heuristic force field for the coupling of the fluid and flexible objects, modeled via Smoothed Particle Hydrodynamics (SPH) and Absolute Nodal Coordinate formulation (ANCF), respectively. The suggested force field, which involves some heuristic parameters to enforce the fluid–solid coupling, cannot approximate the FSI interaction at a resolution finer than that of the fluid discretization. In this sense, it is equivalent to all other approaches proposed for the implementation of wall boundary condition with the caveats that: (1) the procedure of finding the minimum distance between fluid markers and solid surfaces can be prohibitively tedious, particularly for complex shapes; and (2) the wrong choice of heuristic parameters at a certain flow condition can result in either an inexact coupling or a stiff force model which can lead to numerical instability. Additionally, little, if any, is said about the performance of the co-simulation approach. Similarly, Hu et al. [22] approached the FSI problem using SPH and ANCF; however, they implemented the method of moving boundary to couple the fluid dynamics to solid objects.

This contribution is a further development to FSI simulation approaches presented in [22, 38, 39, 42] and also includes a moving boundary approach for two-way fluid–solid coupling implemented through the use of so-called Boundary Condition Enforcing (BCE) markers. Neither Schörgenhumer et al. [42] nor Hu et al. [22] addressed the solid–solid interaction required for many-body FSI problems. In the present work, support for many-body FSI problems, such as those encountered in suspension and polymer flow, is provided by incorporating a lubrication force model. In addition, we have continued our previous validation efforts by benchmarking the dynamics of flexible bodies against that of rigid objects, a study which links the validation of flexible bodies to that of rigid bodies presented in [39]. Finally, we provide a high performance implementation that leverages parallel computing on Graphical Processing Unit (GPU) cards. A complete scaling and time analysis performed herein demonstrate a typical ten-fold speedup compared to the results provided in [22] for problems of comparable size.

The remainder of this document is organized as follows. The various algorithmic components of the proposed simulation framework are discussed in Sect. 2.2, with details on their high performance computing implementation provided in Sect. 2.3. We provide simulation results in Sect. 2.4, including validation and parametric studies, and conclude with some final remarks in Sect. 2.5.

2.2 Simulation Methodology

The simulation framework developed herein relies on: (i) SPH for the simulation of fluid flow, (ii) Newton–Euler 3D rigid body equations of motion, and (iii) ANCF to capture the dynamics of deformable objects. The remainder of this section describes in more details each of these algorithmic components, including a discussion on the formulation adopted for fluid–solid interaction through BCE markers in Sect. 2.2.4

and the methodology used for short range solid–solid interaction through a lubrication force model in Sect. 2.2.5.

2.2.1 The Smoothed Particle Hydrodynamics Method

SPH Liu and Liu [29], Monaghan [31, 34] is a Lagrangian method that probes the fluid domain at a set of moving markers. Each marker has an associated kernel function $W(\mathbf{r}, h)$ defined over a support domain $S(h)$, where $\mathbf{r}$ is the distance from the SPH marker and h is a characteristic length that defines the kernel smoothness. The kernel function should converge to the Dirac delta function as the size of the support domain tends to zero: $\lim_{h\to 0} W(\mathbf{r}, h) = \delta(\mathbf{r})$, be symmetric: $W(\mathbf{r}, h) = W(-\mathbf{r}, h)$, and normal: $\int_S W(\mathbf{r}, h)\mathrm{d}V = 1$, where $\mathrm{d}V$ denote the differential volume. Based on the aforementioned properties, an SPH spatial discretization results in a second order numerical method. Kernel functions must satisfy additional properties [29]; most importantly, they should be positive and monotonically decreasing functions of $\mathbf{r}$. In addition, for computational efficiency, it is advantageous to only consider kernel functions with compact support. A typical kernel function, used throughout this work, is the standard cubic spline kernel, defined as:

$$W(q,h) = \frac{1}{4\pi h^3} \times \begin{cases} (2-q)^3 - 4(1-q)^3, & 0 \le q < 1 \\ (2-q)^3, & 1 \le q < 2, \\ 0, & q \ge 2 \end{cases} \tag{2.1}$$

where $q = |\mathbf{r}|/h$. In general, the radius of the support domain, κh (see Fig. 2.1), is proportional to the characteristic length h, with $\kappa = 2$ for the kernel function of Eq. (2.1).

With ρ and μ denoting the fluid density and viscosity, $\mathbf{v}$ and p the flow velocity and pressure, and m the mass associated with an SPH marker, the continuity equation

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}, \tag{2.2}$$

and the momentum equation

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2 \mathbf{v} + \mathbf{f}, \tag{2.3}$$

are discretized within the SPH framework as [35]:

$$\frac{d\rho_a}{dt} = \rho_a \sum_b \frac{m_b}{\rho_b} (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla_a W_{ab}, \tag{2.4}$$

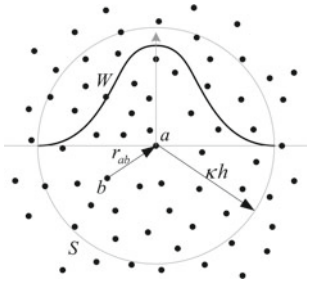


Fig. 2.1 Illustration of the kernel, W, and support domain, S. SPH markers are shown as *black dots*. For 2D problems the support domain is a *circle*, while for 3D problems it is a *sphere*

and

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\left(\frac{p_a}{{\rho_a}^2} + \frac{p_b}{{\rho_b}^2} \right) \nabla_a W_{ab} + \Pi_{ab} \right) + \mathbf{f}_a. \tag{2.5}$$

In Eq. (2.5), indices a and b denote the SPH markers, as shown in Fig. 2.1, and

$$\Pi_{ab} = -\frac{(\mu_a + \mu_b)\mathbf{r}_{ab} \cdot \nabla_a W_{ab}}{\bar{\rho}_{ab}^2 (r_{ab}^2 + \varepsilon \bar{h}_{ab}^2)} \mathbf{v}_{ab} \tag{2.6}$$

imposes the viscous force based on the discretization of the ∇^2 operator, where ε is a regularization coefficient. Here ∇_a indicates the gradient with respect to $\mathbf{r}_a$, i.e. $\partial/\partial \mathbf{r}_a$. Quantities with an over-bar are the average of the corresponding quantities for markers a and b. Summations in the above equations are over all markers within the support domain of marker a. We have evaluated several definitions for the viscosity, as well as different discretizations of ∇^2 [34, 35] in conjunction with simulation of transient Poiseuille flow and concluded that Π_{ab} of Eq. (2.6) leads to the most accurate results for the widest range of Reynolds numbers. It is also worth noting that Eq. (2.6) makes use of the physical fluid viscosity, unlike the use of tuning parameters in artificial viscosity formulations [34].

The pressure p is evaluated using an equation of state [34]:

$$p = \frac{c_s^2 \rho_0}{\gamma} \left\{ \left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right\}, \tag{2.7}$$

where ρ_0 is the fluid reference density, γ is a parameter controlling the stiffness of the pressure-density relationship, and c_s is the speed of sound. In the weakly compressible SPH method, c_s is adjusted based on the maximum speed of the flow, $V_{\max}$, to keep the flow compressibility below any arbitrary value. We chose $\gamma = 7$ and $c_s = 10 \cdot V_{\max}$, which allows 1 % flow compressibility [34]. The fluid flow Eqs. (2.4) and (2.5) are solved in conjunction with the kinematic equation

$$\frac{d\mathbf{r}_a}{dt} = \mathbf{v}_a \tag{2.8}$$

to update the positions of all SPH markers.

Compared to Eq. (2.4), which evaluates the time derivative of the density, the original SPH summation formula calculates the density according to

$$\rho_a = \sum_b m_b W_{ab}. \tag{2.9}$$

Equation (2.4) was preferred to Eq. (2.9) since it produced a smooth density field and worked well for markers close to the boundaries, namely free surfaces, solid interfaces, and wall boundaries. However, Eq. (2.4) does not guarantee consistency between density at a marker and the associated mass and volume [6, 33, 35]. On the other hand, using Eq. (2.9) has problems of its own, in particular large variations in the density field, especially close to the boundary. One of the approaches suggested to resolve this issue is to combine the two methods in a so-called "density re-initialization technique" [9] in which Eq. (2.4) is enforced at each time step while Eq. (2.9) is used to correct any mass-density inconsistencies every n time steps. The results reported herein were obtained with $n = 10$. The Moving Least Squares method or a normalized version of Eq. (2.9) are alternative solutions to the aforementioned issues [9, 11].

Finally, to prevent extensive overlap of marker support domains and enhance incompressibility of the flow, we employ the extended SPH approach (XSPH) as described in [32]. The XSPH correction takes into account the velocity of neighboring markers through a mean velocity evaluated within the support of a nominal marker a as

$$\langle \mathbf{v}_a \rangle = \mathbf{v}_a + \Delta \mathbf{v}_a, \tag{2.10}$$

where

$$\Delta \mathbf{v}_a = \zeta \sum_b \frac{m_b}{\bar{\rho}_{ab}} (\mathbf{v}_b - \mathbf{v}_a) W_{ab} \tag{2.11}$$

and $0 \leq \zeta \leq 1$ adjusts the contribution of velocities of neighboring markers. All simulations presented in this work were obtained with $\zeta = 0.5$. The modified velocity calculated from Eq. (2.10) replaces the original velocity in the density and position update equations, but not in the momentum equation [9].

2.2.2 Rigid Body Dynamics

The dynamics of rigid bodies is fully characterized by the Newton–Euler equations of motion (EOM), see for instance [18]. For each body $i = 1, 2, \ldots, n_b$ present in the system, we have:

$$\frac{d\mathbf{V}_i}{dt} = \frac{\mathbf{F}_i}{M_i}, \tag{2.12}$$

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{V}_i, \tag{2.13}$$

$$\frac{d\boldsymbol{\omega}_i'}{dt} = \mathbf{J}_i'^{-1}\left(\mathbf{T}_i' - \tilde{\boldsymbol{\omega}}_i'\mathbf{J}_i'\boldsymbol{\omega}_i'\right), \tag{2.14}$$

$$\frac{d\mathbf{q}_i}{dt} = \frac{1}{2}\mathbf{G}_i^T\boldsymbol{\omega}_i', \tag{2.15}$$

and

$$\mathbf{q}_i^T\mathbf{q}_i - 1 = 0, \tag{2.16}$$

where $\mathbf{F}_i$ and $\mathbf{T}_i'$ represent the external forces and torques acting on body i, including fluid–solid interaction forces obtained as described in Sect. 2.2.4. The quantities $\mathbf{X}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^4$ denote the position vector and rotation quaternion, while $\mathbf{V}_i$, $\boldsymbol{\omega}_i' \in \mathbb{R}^3$ represent the linear and angular body velocities. The mass and moment of inertia are denoted by M_i and $\mathbf{J}_i'$, respectively. Quantities with a prime symbol are represented in the rigid body local reference frame. Given $\mathbf{a} = \left[a_x,\ a_y,\ a_z\right]^T \in \mathbb{R}^3$ and $\mathbf{q} = \left[q_x,\ q_y,\ q_z,\ q_w\right]^T \in \mathbb{R}^4$, the auxiliary matrices $\tilde{\mathbf{a}}$ and $\mathbf{G}$ are defined as:

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} -q_y & q_x & q_w & -q_z \\ -q_z & -q_w & q_x & q_y \\ -q_w & q_z & -q_y & q_x \end{bmatrix}. \tag{2.17}$$

2.2.3 *Flexible Body Dynamics*

For the simulation of flexible solid bodies suspended in the fluid, we adopt the ANCF formulation [44] which allows for large deformations and large rigid body rotations. While extension to other elastic elements is straightforward, in the current Chrono::Fluid implementation we only support gradient deficient ANCF beam elements which are used to model slender flexible bodies composed of n_e adjacent ANCF beam elements. In this approach, we model the flexible bodies using a number $n_n = n_e + 1$ of equally-spaced node beam elements, each represented by 6 coordinates, $\mathbf{e}_j = [\mathbf{r}_j^T,\ \mathbf{r}_{j,x}^T]^T$, $j = 0, 1, \ldots, n_e$, representing the three components of the global position vector of the node and the three components of the position vector gradient. This is therefore equivalent to a model using n_e ANCF beam elements with $6 \times n_n$ continuity constraints, but is more efficient in that it uses a minimal set of coordinates. We note that formulations using gradient deficient ANCF beam elements display no shear locking problems [15, 43, 45] and, due to the reduced number

of nodal coordinates, are more efficient than fully parameterized ANCF elements. However, gradient deficient ANCF beam elements cannot describe a rotation about its axis and therefore cannot model torsional effects.

Consider first a single ANCF beam element of length ℓ. The global position vector of an arbitrary point on the beam centerline, specified through its element spatial coordinate $0 \leq x \leq \ell$, is then obtained as

$$\mathbf{r}(x, \mathbf{e}) = \mathbf{S}(x)\mathbf{e}, \tag{2.18}$$

where $\mathbf{e} = [\mathbf{e}_l^T,\ \mathbf{e}_r^T]^T \in \mathbb{R}^{12}$ is the vector of element nodal coordinates. With $\mathbf{I}$ being the 3×3 identity matrix, the 3×12 shape function matrix $\mathbf{S} = [S_1\mathbf{I} \quad S_2\mathbf{I} \quad S_3\mathbf{I} \quad S_4\mathbf{I}]$ is defined using the shape functions [44]

$$\begin{aligned} S_1 &= 1 - 3\xi^2 + 2\xi^3 \\ S_2 &= \ell\left(\xi - 2\xi^2 + \xi^3\right) \\ S_3 &= 3\xi^2 - 2\xi^3 \\ S_4 &= \ell\left(-\xi^2 + \xi^3\right), \end{aligned} \tag{2.19}$$

where $\xi = x/\ell \in [0, 1]$.

The element EOM are then written as

$$\mathbf{M}\ddot{\mathbf{e}} + \mathbf{Q}^e = \mathbf{Q}^a, \tag{2.20}$$

where $\mathbf{Q}^e$ and $\mathbf{Q}^a$ are the generalized element elastic and applied forces, respectively, and $\mathbf{M} \in \mathbb{R}^{12\times 12}$ is the symmetric consistent element mass matrix defined as

$$\mathbf{M} = \int_\ell \rho_s A \mathbf{S}^T \mathbf{S}\, dx. \tag{2.21}$$

The generalized element elastic forces are obtained from the strain energy expression [44] as

$$\mathbf{Q}^e = \int_\ell EA\varepsilon_{11} \left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}}\right)^T dx + \int_\ell EI\kappa \left(\frac{\partial \kappa}{\partial \mathbf{e}}\right)^T dx, \tag{2.22}$$

where $\varepsilon_{11} = \left(\mathbf{r}_x^T\mathbf{r}_x - 1\right)/2$ is the axial strain and $\kappa = \|\mathbf{r}_x \times \mathbf{r}_{xx}\|/\|\mathbf{r}_x\|^3$ is the magnitude of the curvature vector. The required derivatives of the position vector $\mathbf{r}$ can be easily obtained from Eq. (2.18) in terms of the derivatives of the shape functions as $\mathbf{r}_x(x, \mathbf{e}) = \mathbf{S}_x(x)\mathbf{e}$ and $\mathbf{r}_{xx}(x, \mathbf{e}) = \mathbf{S}_{xx}(x)\mathbf{e}$.

External applied forces, in particular the forces due to the interaction with the fluid (see Sect. 2.2.4), are included as concentrated forces at a BCE marker. The corresponding generalized forces are obtained from the expression of the virtual work as

$$\mathbf{Q}^a = \mathbf{S}^T(x_a)\mathbf{F}, \tag{2.23}$$

where $\mathbf{F}$ is the external point force and the shape function matrix is evaluated at the projection onto the element's centerline of the force application point. If considered, the generalized gravitational force can be computed as

$$\mathbf{Q}^g = \int_\ell \rho_s A \mathbf{S}^T \mathbf{g}\, dx. \tag{2.24}$$

In the above expressions, ρ_s represents the element mass density, A is the cross section area, E is the modulus of elasticity, and I is the second moment of area.

The EOM for a slender flexible body composed of n_e ANCF beam elements are obtained by assembling the elemental EOMs of Eq. (2.20) and taking into consideration that adjacent beam elements share 6 nodal coordinates. Let $\hat{\mathbf{e}} = [\mathbf{e}_0^T,\ \mathbf{e}_1^T, \ldots\ \mathbf{e}_{n_e}^T]^T$ be the set of independent nodal coordinates; then the nodal coordinates for the jth element can be written using the mapping

$$\begin{bmatrix}\mathbf{e}_l \\ \mathbf{e}_r\end{bmatrix}_j = \mathbf{B}_j\hat{\mathbf{e}}, \quad \text{with } \mathbf{B}_j = \begin{bmatrix}\mathbf{0}\ \mathbf{0}\ \ldots \mathbf{I}_3\ \mathbf{0}\ \ldots \mathbf{0} \\ \mathbf{0}\ \mathbf{0}\ \ldots \mathbf{0}\ \mathbf{I}_3\ \ldots \mathbf{0}\end{bmatrix} \tag{2.25}$$

and the assembled EOMs are obtained, from the principle of virtual work, as follows. Denoting by $\mathbf{M}_j$ be the element mass matrix of Eq. (2.21) for the jth ANCF beam element, it can be written in block form as

$$\mathbf{M}_j = \begin{bmatrix}\mathbf{M}_{j,ll} & \mathbf{M}_{j,lr} \\ \mathbf{M}_{j,rl} & \mathbf{M}_{j,rr}\end{bmatrix}, \tag{2.26}$$

where $\mathbf{M}_{j,lr} = \mathbf{M}_{j,rl}^T$ and all sub-blocks have dimension 6×6. Here, l denotes the *left* end of the beam element, i.e., the node characterized by the nodal coordinates $\mathbf{e}_{j-1}$, while r corresponds to the node with coordinates $\mathbf{e}_j$. With a similar decomposition of a generalized element force into

$$\mathbf{Q}_j = \begin{bmatrix}\mathbf{Q}_{j,l} \\ \mathbf{Q}_{j,r}\end{bmatrix} \tag{2.27}$$

we obtain

$$\hat{\mathbf{M}}\ddot{\hat{\mathbf{e}}} = \hat{\mathbf{Q}}^a - \hat{\mathbf{Q}}^e \tag{2.28}$$

where

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_{1,ll} & \mathbf{M}_{1,lr} & & & & \\ \mathbf{M}_{1,rl} & \mathbf{M}_{1,rr} + \mathbf{M}_{2,ll} & \mathbf{M}_{2,lr} & & & \\ & \mathbf{M}_{2,rl} & \mathbf{M}_{2,rr} + \mathbf{M}_{3,ll} & & & \\ & & & \ddots & & \\ & & & & \mathbf{M}_{n_e,rr} \end{bmatrix} \tag{2.29}$$

$$\hat{\mathbf{Q}}^a - \hat{\mathbf{Q}}^e = \begin{bmatrix} \sum \mathbf{Q}^a_{1,l} \\ \sum \mathbf{Q}^a_{1,r} + \sum \mathbf{Q}^a_{2,l} \\ \sum \mathbf{Q}^a_{2,r} + \sum \mathbf{Q}^a_{3,l} \\ \vdots \\ \sum \mathbf{Q}^a_{n_e,r} \end{bmatrix} - \begin{bmatrix} \mathbf{Q}^e_{1,l} \\ \mathbf{Q}^e_{1,r} + \mathbf{Q}^e_{2,l} \\ \mathbf{Q}^e_{2,r} + \mathbf{Q}^e_{3,l} \\ \vdots \\ \mathbf{Q}^e_{n_e,r} \end{bmatrix}. \tag{2.30}$$

Finally, we note that inclusion of additional constraints (e.g., anchoring the beam at one end to obtain a flexible cantilever or fixing its position only to obtain a flexible pendulum) can be done either by formulating the EOM as differential-algebraic equations or by deriving an underlying ODE after explicitly eliminating the corresponding constrained nodal coordinates. The latter approach was used in all simulations involving flexible cantilevers that are discussed in Sect. 2.4.

2.2.4 Fluid–Solid Interaction

The two-way fluid–solid coupling was implemented based on a methodology described in [38]. The state update of any SPH marker relies on the properties of its neighbors and resolves shear as well as normal inter-marker forces. For the SPH markers close to solid surfaces, the SPH summations presented in Eqs. (2.4), (2.5), (2.9), and (2.11) capture the contribution of fluid markers. The contribution of solid objects is calculated using BCE markers placed on and close to the solid surface as shown in Fig. 2.2. In the case of flexible beams, the BCE markers are placed on "rigid disks" that are uniformly-spaced along the beam's axis and whose normals always coincide with the local tangent to the beam's axis. In all cases, the BCE marker locations are initialized so that the distance between two neighboring BCE markers is approximately equal to the initial distance between two SPH markers; in particular, this is also the distance between two adjacent disks of BCE markers in Fig. 2.2b.

The velocity of a BCE marker is obtained from the rigid/deformable body motion of the solid and as such ensures the no-slip condition on the solid surface. Including the BCE markers in the SPH summation Eqs. (2.4) and (2.5) thus enforces the solid-to-fluid coupling. On the other hand, fluid-to-solid coupling is realized by applying the quantity in the right-hand side of Eq. (2.5), evaluated at each BCE marker, as an external force on the corresponding rigid or deformable solid body using Eqs. (2.12) and (2.23), respectively.

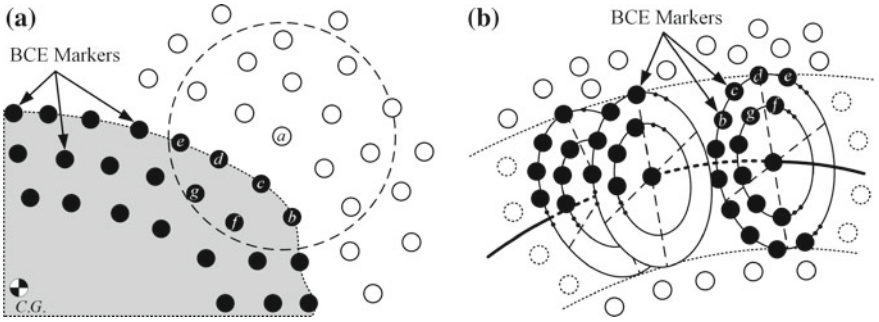


Fig. 2.2 Fluid–solid interaction using BCE markers attached to a body: (**a**) rigid body; (**b**) flexible beam. BCE and fluid markers are represented by *black* and *white circles*, respectively. The BCE markers positioned in the interior of the body should be placed to a depth no larger than the size of the compact support S of the kernel W

2.2.5 Short Range Interaction

Dry friction models typically used to characterize the dynamics of granular materials [3, 23, 24] do not resolve the impact of solid surfaces in hydrodynamics media. In practice, it is unfeasible to resolve the short-range, high-intensity impact forces in wet media due to the computational limits on space and time resolution. In reality, particle boundaries are not smooth and physical contact can happen [20]. By assuming smooth surfaces, Davis et al. followed the Hertz contact theory of linear elasticity to calculate the pressure at the interface of two approaching elastic spheres in close proximity [10]. Their calculation showed that particles do not rebound at small Stokes number, $St = (2/9)(\rho_s/\rho)Re_p$, where ρ_s and Re_p are the solid particle density and particle Reynolds number, respectively. The minimum St for a rebound after the hydroelastic impact depends on the spheres' rigidity. For rigid spheres, rebound happens at $St > 10$. An alternative approach to calculate the singular forces at contact relies on lubrication theory [13]. Ladd [26] proposed a normal lubrication force between two spheres that increases rapidly as the distance between particles approaches zero and therefore prevents the actual touching of the spheres:

$$\mathbf{F}_{ij}^{lub} = \min\left\{-6\pi\mu a_{ij}^2\left(\frac{1}{s}-\frac{1}{\Delta_c}\right), 0\right\}\cdot \mathbf{v}_{n_{ij}}, \quad \text{where } \frac{1}{a_{ij}} = \frac{1}{a_i}+\frac{1}{a_j}. \tag{2.31}$$

Here, a_i and a_j are the sphere radii, $\mathbf{v}_{n_{ij}}$ is the normal component of the relative velocity, and s is the distance between surfaces. For $s > \Delta_c$, $\mathbf{F}_{ij}^{lub} = 0$ and the spheres are subject only to hydrodynamic forces. Ladd and Verberg [27] demonstrated good agreement of the proposed lubrication force with Brenner's exact solution [7].

Equation (2.31) provides a simplistic model for the estimation of the lubrication force in normal direction. Generalization of this model to non-spherical objects requires the calculation of the minimum distance and curvature of the

contact surfaces. By adopting the approach proposed in [12] for lubrication force in lattice Boltzmann method, we calculate the partial lubrication force by modifying Eq. (2.31) as

$$\mathbf{F}_{ij}^{lub} = \sum_{k} \mathbf{f}_{ij}^{k}, \quad \text{with } \mathbf{f}_{ij}^{k} = \min\left\{-\frac{3}{2}\pi\mu h^2\left(\frac{1}{s^*} - \frac{1}{\Delta_c}\right), 0\right\} \cdot \mathbf{v}_{n_{ij}}^{*}, \tag{2.32}$$

where s^* and $\mathbf{v}_{n_{ij}}^{*}$ denote the markers relative distance and velocity, respectively, and the summation is over all interacting markers of two solid objects.

2.3 GPU-Based Implementation

Chrono::Fluid [8], an open-source simulation framework for fluid–solid interaction, relies on a second order explicit Runge-Kutta method [5] for time integration of fluid, rigid, and flexible bodies, and a parallel implementation of the spatial subdivision method on the GPU for construction of the markers neighbor lists. In what follows, the computation kernels and their implementations are described with more details.

At the beginning of each time step, a neighbor list is assembled to indicate the set of markers that fall within the kernel support of each marker; if N markers are used in the simulation, N lists are generated. The force components appearing on the right hand side of Eqs. (2.4), (2.5), and (2.31) are subsequently computed based on these neighbor lists. Two different functions are called to capture the interaction between markers according to their types; i.e., fluid or solid, via SPH or the short range interaction model described in Sect. 2.2.5. In the second stage, the state of the fluid markers, including position, velocity, and density, is updated based on Eqs. (2.4), (2.5), and (2.8). The state of each rigid body is updated according to Eqs. (2.12) through (2.15). This is followed by time integration of deformable body motion according to Eq. (2.28). Since a rigid wall boundary is a particular instance of a rigid body (with zero or other predefined velocity), it requires no special treatment.

Stable integration of the SPH fluid equations requires step-sizes which are also appropriate for propagating the dynamics of any rigid solids in the FSI system. However, integration of the dynamics of deformable bodies, especially as their stiffness increases, may require smaller time steps. To accommodate this requirement, while minimizing any adverse effects on the overall simulation efficiency, we have implemented a simple dual-rate integration scheme using intermediate steps for the integration of the flexible dynamics EOMs (typically $\Delta t_{SPH}/\Delta t_{ANCF} = 10$, although stiffer problems may require ratios of up to 50). We note that typical FSI simulation models involve a number of SPH markers many orders of magnitude larger than that of ANCF nodal coordinates required for the flexible bodies. As such, the execution time required for integration of the flexible body dynamics is a negligible fraction of the computation time for propagating the SPH equations and therefore the dual-rate integration scheme has no effect on the net overall simulation efficiency.

The above algorithm was implemented to execute in parallel on GPU cards using CUDA [37]. The hardware used to run the simulations that produced the results reported in this contribution, NVIDIA Kepler K20X, has 2688 parallel scalar processors. At each time step, five different tasks are executed on the GPU to (1) calculate the inter-marker forces, (2) carry out fluid time integration, (3) carry out rigid body time integration, (4) carry out deformable body time integration, and (5) enforce boundary conditions. The lists of neighbors needed to evaluate the inter-marker forces are generated via a proximity computation algorithm based on a decomposition of the computation domain into cubic bins. The side length of each bin is roughly equal to the size of the support domain of an SPH marker. A hash table is used to sort the markers according to their location in the domain. Based on the sorted hash table, each marker accesses the list of markers intersecting its own and neighboring bins to calculate the forcing terms. The proximity computation algorithm uses the parallel sorting and scan algorithms provided by the Thrust library [19].

To improve the code vectorization through coalesced memory access and use of fast memory (L1/L2 cache, shared memory, and registers), each computation task was implemented as a sequence of light-weight GPU kernels. For instance, different computation kernels are implemented to update the attributes of the solid bodies, including force, moment, rotation, translation, linear and angular velocity, and location of the BCE markers. A similar coding style was maintained for the density re-initialization, boundary condition implementation, and mapping of the markers' data on an Eulerian grid for post processing.

2.4 Results and Discussion

The robustness and accuracy of the fluid flow and coupled fluid-rigid body simulation was demonstrated in previous work. See [39] for a comprehensive set of validation studies of rigid particle migration and suspension distribution in pipe flow. Herein, we focus on recent extensions to Chrono::Fluid to support fluid-deformable body interaction and present additional numerical experiments to validate the flexible body simulation algorithm , as well as several simulation-based studies involving coupling of fluid flow and deformable bodies.

The simulations presented in this section involve relatively soft beams (with a modulus of elasticity $E \leq 20$ MPa) that are either unconstrained or else anchored at one end. Since computational efficiency of the FSI code is directly related to the number of nodal coordinates used to model the flexible beams, we first conducted a parametric study to identify the minimum number of ANCF beam elements required to accurately capture the dynamics of interest in the subsequent experiments. In this set of experiments, we considered a cantilever of length $L = 1$ m and diameter $d = 0.04$ m with density $\rho_s = 7{,}200\,\mathrm{kg/m^3}$ and modulus of elasticity $E = 20$ Mpa under gravity ($g = -9.81\ \mathrm{m/s^2}$) in vacuum or immersed in fluid of various viscosities. Simulation results using different number of ANCF beam elements ($n_e = 2, 3, 4, 5$) showed acceptable convergence at all discretizations and virtually identical results

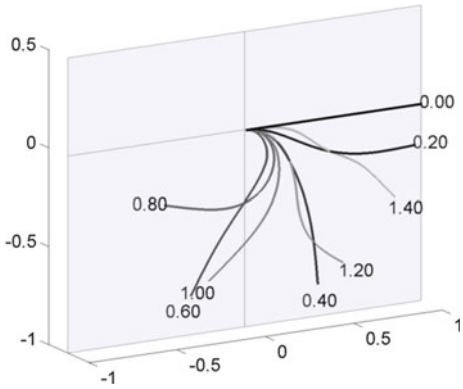


Fig. 2.3 Time snapshots of a flexible cantilever moving under the action of gravity (in vacuum). The *darker colors* denote earlier stages of the motion

for $n_e \geq 4$. Figure 2.3 shows a few time snapshots from a dynamic simulation of a cantilever modeled with $n_e = 4$ ANCF beam elements, the value which was selected for all subsequent simulations.

2.4.1 Floating Beam in Poiseuille Flow

Ongoing work is aimed at validating the fluid-deformable solid interaction code against experimental and analytical results [40]. Here we present a comparison against the already validated fluid-rigid solid simulation code. For this purpose we conducted a series of numerical experiments involving short stiff deformable beams and equivalent rigid cylinders free floating in channel Poiseuille flow.

The validation test was performed using a straight beam with $L = 0.2$ m, $\rho_s = 7{,}200$ kg/m^3, $E = 20$ MPa, $d = 0.04$ m and a rigid cylinder with the same density and geometry. The beam and rigid cylinder were subjected to an accelerating channel flow aligned with the global x axis with final steady state Reynolds number $Re_c = \rho V_{ave} w/\mu = 100$, where $\rho = 1{,}000$ kg/m^3, $\mu = 1$ N s/m^2, average velocity $V_{ave} = 0.2$ m/s, and channel width $w = 1$ m. The beam and cylinder were initially perpendicular to the flow and rotated in the yz plane. Comparisons of the resulting beam orientation angles, relative to the global x, y, and z axes, and of the time evolution of the velocity in the x direction of the beam center velocity are presented in Fig. 2.4. The results show good agreement with differences due to the inability of the gradient deficient ANCF beam element model to capture rotation about the beam's axis.

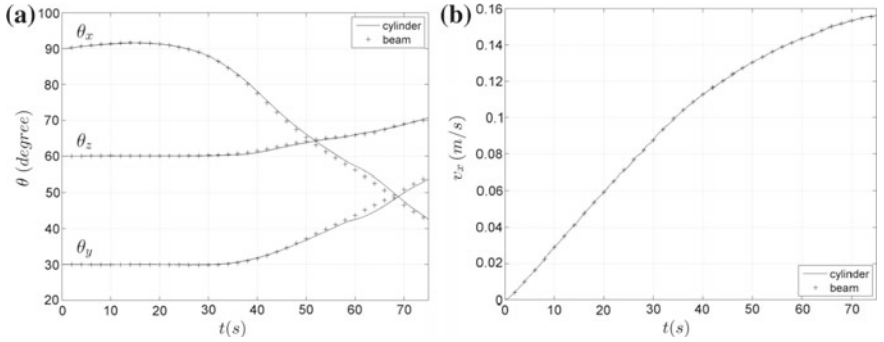


Fig. 2.4 Comparison of the dynamics of a rigid cylinder and of a corresponding stiff deformable beam under accelerating channel flow: (**a**) beam orientation; (**b**) center velocity

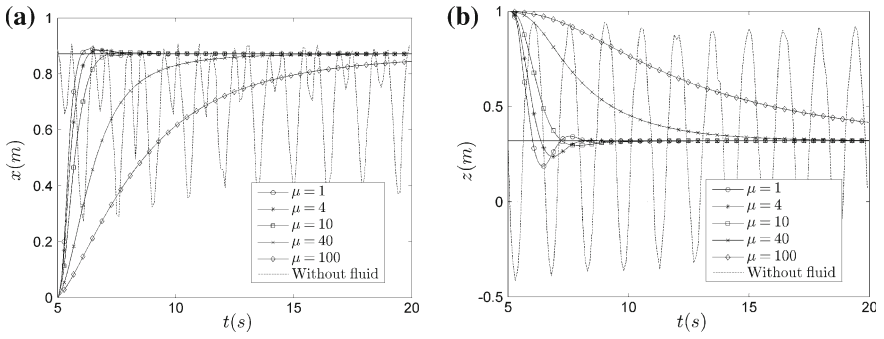


Fig. 2.5 Motion of a cantilever beam in fluid of different viscosities: (**a**) tip displacement in x direction; (**b**) tip displacement in z direction

2.4.2 Flexible Cantilever Immersed in Fluid: Effect of Viscosity

Through a parametric study of the motion of a cantilever moving under the action of gravity in viscous fluid, we investigated the effect of viscosity on the motion of the beam's tip. As shown in Fig. 2.5, the beam motion switches from oscillatory to critically damped motion as the viscosity increases. For the beam parameters used in this study, namely $L = 1$ m, $d = 0.04$ m, $\rho_s = 7{,}200$ kg/m^3, and $E = 20$ MPa, the switch between the two behaviors is observed to occur around $\mu \simeq 10$ N s/m^2. It was also noticed that viscosity has little effect on the trajectory of the beam tip (plots are not provided). Nevertheless, compared to the case of a cantilever moving in vacuum, when immersed in fluid, the tip moves on a much shorter path. This deviation, i.e. having the same trajectory regardless of the fluid viscosity, which is different from that of a cantilever in vacuum, is most probably due to the pressure drag which is added to the viscous drag considered herein.

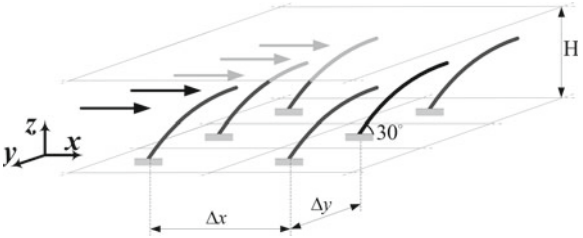


Fig. 2.6 Arrays of flexible cantilever beams in laminar channel flow. The beams, laid out in an uniform grid, are anchored at an angle of 30° in the direction of the flow

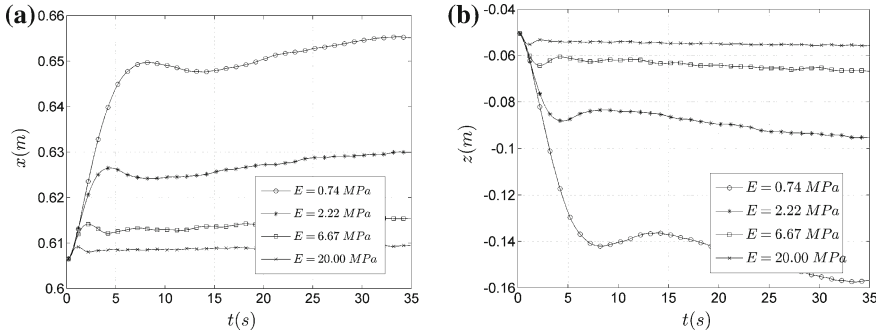


Fig. 2.7 Motion of a cantilever beam of different elasticity modulus in laminar channel flow: (**a**) tip displacement in x direction; (**b**) tip displacement in z direction

2.4.3 Impulsively Started Motion of Cantilevers in Channel Flow: Effect of Elasticity

Vibration behavior of flexible beams in viscous fluid was studied by considering an array of cantilevers in channel flow. Unlike the test described in Sect. 2.4.2, here the flexible cantilevers are initially at rest when they are hit by a laminar channel flow. This model can be used to study the effect of horizontal waves on beams submerged in a fluid.

The array of flexible cantilevers is laid out in the xy plane, with $(\Delta x, \Delta y) = (1.2, 0.4)$ m, as shown in Fig. 2.6, thus allowing interaction of the beams through the flow. Each beam is anchored in the xz plane with an angle of 30° with respect to the y axis. The fluid, with density $\rho = 1{,}000$ kg/m^3 and viscosity $\mu = 1$ N s/m^2, flows in the x direction between two planes spaced by $H = 1$ m vertically.

Figure 2.7 shows the tip deformation of one cantilever beam for different modulus of elasticity in the range $E \in (0.25, 20)$ MPa. All other beam parameters were kept fixed at $L = 0.7$ m, $d = 0.04$ m, and $\rho_s = 7{,}200$ kg/m^3.

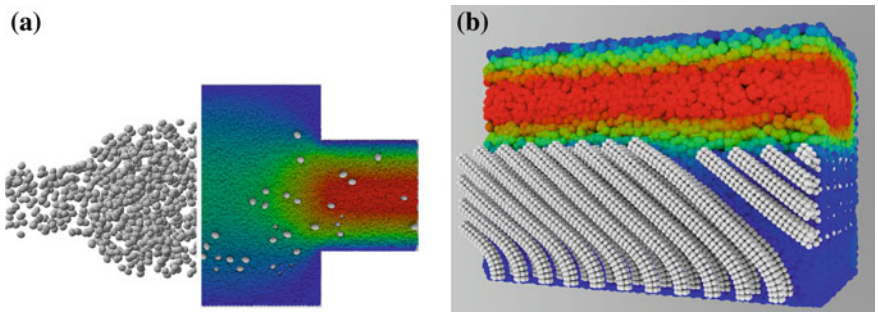


Fig. 2.8 FSI problems considered for scalability analysis: (**a**) flow of a dense suspension of rigid particles through a step pipe; for clarity, the *left half* of the image shows the rigid particles only, while the *right half* shows both rigid particles and SPH markers at the pipe mid-section; (**b**) channel flow over an array of flexible cantilever beams; for visualization purposes only, marker sizes are artificially changed

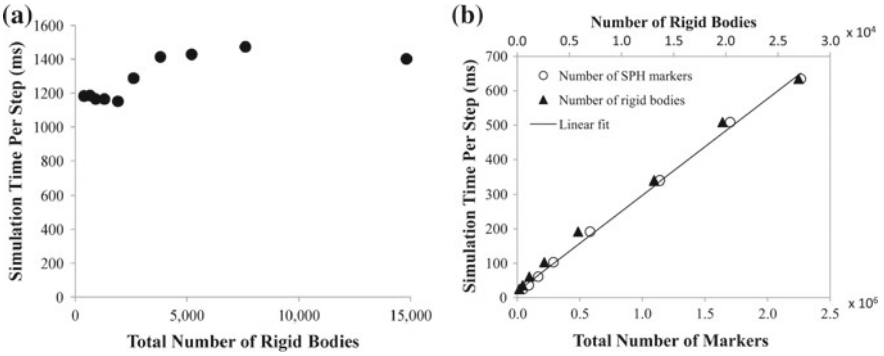


Fig. 2.9 Scaling analysis of Chrono::Fluid for fluid-rigid body interaction problems: (**a**) simulation time versus number of rigid bodies for a total number of 3×10^6 markers; (**b**) simulation time as a function of combined problem size

2.4.4 Scalability Analysis

Scalability of Chrono::Fluid was investigated through simulations of multi-component problems including the flow of flexible and rigid objects in flow, samples of which are provided in Fig. 2.8.

As shown in Fig. 2.9a, an increase in the number of rigid bodies present in the system only marginally affects the total simulation time. This is due to the fact that the number of BCE markers associated with solid bodies is only a very small fraction of the number of SPH discretization markers, the latter dictating to a very large extent the required computation time. We must however mention that, as the concentration of solid objects increases, smaller time steps are required since the probability of short-range, high-frequency interactions increases. The same conclusion can be reached

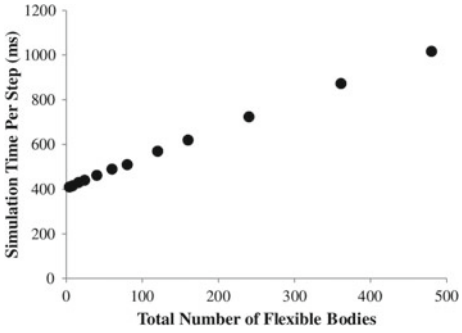


Fig. 2.10 Scaling analysis of Chrono::Fluid for fluid-flexible body interaction problems: simulation time versus number of flexible bodies for a total number of 1.5×10^6 markers

from the results presented in Fig. 2.9b which shows linear growth of the simulation time with the size of the fluid-rigid body mixture problem (i.e., the combined number of SPH markers and rigid bodies).

On the other hand, as seen in Fig. 2.10, we observe only linear scalability when rigid bodies are replaced by flexible beams. This is only a consequence of the current Chrono::Fluid implementation in which the dynamics update for flexible bodies is carried out on the CPU, thus dominating the simulation time as the problem size increases. We expect this will be rectified once this stage of the simulation is also moved to the GPU.

2.5 Conclusions and Future Work

We describe a Lagrangian–Lagrangian approach for the direct numerical simulation of two-way coupled fluid–solid interaction. Building up on previous work [39], the simulation framework Chrono::Fluid was extended beyond fluid-rigid interaction to include deformable solids. For simulations of solid bodies immersed in fluid, the proposed method employs a lubrication force model for incorporating solid–solid interaction and, in the case of deformable bodies, self-contact. We describe simulation results for free-floating flexible beams in Poiseuille flow and channel flow over a grid of flexible cantilevers, and provide parametric studies of the effect of fluid viscosity and material elasticity. These results suggest that the adopted approach has good predictive capabilities and is able to capture the dynamics of the systems under consideration. Moreover, the Lagrangian–Lagrangian formulation is amenable to efficient implementation on GPU cards as indicated by the scalability studies presented herein.

Current effort is aimed at providing a GPU-only implementation, by also parallelizing the flexible body dynamics calculations and updates, with ongoing work focused on additional validation studies using both analytical and experimental data. In addition, we plan on extending the formulation to support 2D and fully-3D deformable solids modeled with ANCF.

Finally, we note that additional examples of Chrono::Fluid simulations can be found at [41]

Acknowledgments Financial support was provided in part by the National Science Foundation (grant NSF CMMI-084044). Support for the second and third authors was provided in part by Army Research Office awards W911NF-11-0327 and W911NF-12-1-0395. NVIDIA is acknowledged for providing the GPU hardware used in generating the simulation results reported herein.

References

1. Amini Y, Emdad H, Farid M (2011) A new model to solve fluid-hypo-elastic solid interaction using the smoothed particle hydrodynamics (SPH) method. Eur J Mech B Fluids 30(2):184–194
2. Andrews M, O'rourke P (1996) The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. Int J Multiph Flow 22(2):379–402
3. Anitescu M, Hart GD (2004) A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. Int J Numer Meth Eng 60(14):2335–2371
4. Antoci C, Gallati M, Sibilla S (2007) Numerical simulation of fluid-structure interaction by SPH. Comput Struct 85(11):879–890
5. Atkinson K (1989) An introduction to numerical analysis. Wiley, USA
6. Benz W (1986) Smoothed particle hydrodynamics: a review. In: Proceedings of the NATO advanced research workshop on the numerical modelling of nonlinear stellar pulsations problems and prospects, Les Arcs, France, 20–24 Mar. Kluwer Academic Publishers, Berlin
7. Brenner H (1961) The slow motion of a sphere through a viscous fluid towards a plane surface. Chem Eng Sci 16(3):242–251
8. Chrono::Fluid (2014) An open source engine for fluid. Solid interaction. http://armanpazouki.github.io/chrono-fluid
9. Colagrossi A, Landrini M (2003) Numerical simulation of interfacial flows by smoothed particle hydrodynamics. J Comput Phys 191(2):448–475
10. Davis RH, Serayssol JM, Hinch E (1986) Elastohydrodynamic collision of two spheres. J Fluid Mech 163:479–497
11. Dilts G (1999) Moving-least-squares-particle hydrodynamics-I. Consistency and stability. Int J Numer Methods Eng 44(8):1115–1155
12. Ding EJ, Aidun CK (2003) Extension of the lattice-Boltzmann method for direct simulation of suspended particles near contact. J Stat Phys 112(3–4):685–708
13. Durlofsky L, Brady JF, Bossis G (1987) Dynamic simulation of hydrodynamically interacting particles. J Fluid Mech 180(1):21–49
14. Fan LS, Zhu C (2005) Principles of gas-solid flows. Cambridge University Press, Cambridge
15. Gerstmayr J, Shabana A (2006) Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. Nonlinear Dyn 45(1):109–130
16. Gidaspow D (1994) Multiphase flow and fluidization: continuum and kinetic theory descriptions. Academic Press, Boston
17. Glowinski R, Pan T, Hesla T, Joseph D (1999) A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int J Multiph Flow 25(5):755–794
18. Haug E (1989) Computer aided kinematics and dynamics of mechanical systems. Allyn and Bacon, Boston
19. Hoberock J, Bell N Thrust: C++ template library for CUDA. http://thrust.github.com/
20. Hocking L (1973) The effect of slip on the motion of a sphere close to a wall and of two adjacent spheres. J Eng Math 7(3):207–221
21. Hu H, Patankar N, Zhu M (2001) Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian–Eulerian technique. J Comput Phys 169(2):427–462

22. Hu W, Tian Q, Hu H (2013) Dynamic simulation of liquid-filled flexible multibody systems via absolute nodal coordinate formulation and SPH method. Nonlinear Dyn, pp 1–19
23. Kruggel-Emden H, Simsek E, Rickelt S, Wirtz S, Scherer V (2007) Review and extension of normal force models for the discrete element method. Powder Technol 171(3):157–173
24. Kruggel-Emden H, Wirtz S, Scherer V (2008) A study on tangential force laws applicable to the discrete element method (DEM) for materials with viscoelastic or plastic behavior. Chem Eng Sci 63(6):1523–1541
25. Ladd AJ (1994) Numerical simulations of particulate suspensions via a discretized Boltzmann equation. J Fluid Mech 271(1):285–339
26. Ladd AJ (1997) Sedimentation of homogeneous suspensions of non-brownian spheres. Phys Fluids 9:491
27. Ladd A, Verberg R (2001) Lattice-Boltzmann simulations of particle-fluid suspensions. J Stat Phys 104(5–6):1191–1251
28. Lee CJK, Noguchi H, Koshizuka S (2007) Fluid-shell structure interaction analysis by coupled particle and finite element method. Comput Struct 85(11):688–697
29. Liu M, Liu G (2010) Smoothed particle hydrodynamics (SPH): an overview and recent developments. Arch Computat Methods Eng 17(1):25–76
30. McLaughlin J (1994) Numerical computation of particles-turbulence interaction. Int J Multiph Flow 20:211–232
31. Monaghan JJ (1988) An introduction to SPH. Comput Phys Comm 48(1):89–96
32. Monaghan J (1989) On the problem of penetration in particle methods. J Comput Phys 82(1):1–15
33. Monaghan JJ (1992) Smoothed particle hydrodynamics. Ann Rev Astron Astrophys 30:543–574
34. Monaghan J (2005) Smoothed particle hydrodynamics. Rep Prog Phys 68(1):1703–1759
35. Morris J, Fox P, Zhu Y (1997) Modeling low Reynolds number incompressible flows using SPH. J Computat Phys 136(1):214–226
36. Negrut D, Tasora A, Mazhar H, Heyn T, Hahn P (2012) Leveraging parallel computing in multibody dynamics. Multibody Sys Dyn 27(1):95–117
37. NVIDIA: CUDA developer zone (2014). https://developer.nvidia.com/cuda-downloads
38. Pazouki A, Negrut D (2012) Direct simulation of lateral migration of buoyant particles in channel flow using GPU computing. In: Proceedings of the 32nd computers and information in engineering conference, CIE32, 12–15 Aug 2012. American Society of Mechanical Engineers, Chicago, IL, USA
39. Pazouki A, Negrut D A numerical study of the effect of particle properties on the radial distribution of suspensions in pipe flow. Submitted to International Journal of Multiphase Flow
40. Sader J (1998) Frequency response of cantilever beams immersed in viscous fluids with applications to the atomic force microscope. J Appl Phys 84(1):64–76
41. SBEL: Vimeo page (2014). https://vimeo.com/uwsbel
42. Schörgenhumer M, Gruber PG, Gerstmayr J (2013) Interaction of flexible multibody systems with fluids analyzed by means of smoothed particle hydrodynamics. Multibody Sys Dyn, pp 1–24
43. Schwab A, Meijaard J (2005) Comparison of three-dimensional flexible beam elements for dynamic analysis: finite element method and absolute nodal coordinate formulation. In: Proceedings of the ASME 2005 IDETC/CIE, Orlando, Florida
44. Shabana A (2005) Dynamics of multibody systems. Cambridge University Press, New York
45. Shabana A (1997) Flexible multibody dynamics: review of past and recent developments. Multibody Sys Dyn 1:339–348
46. Zhang D, Prosperetti A (1994) Averaged equations for inviscid disperse two-phase flow. J Fluid Mech 267:185–220

Chapter 3
Strategies for Adaptive Model Reduction with DCA-Based Multibody Modeling of Biopolymers

Jeremy J. Laflin, Kurt S. Anderson and Imad M. Khan

Abstract This contribution discusses the need for adaptive model reduction when simulating biopolymeric systems and the issues surrounding the execution of these model changes in a computationally efficient manner. These systems include nucleic acids, proteins, and traditional polymers such as polyethylene. Two distinct general strategies of reducing selected degrees-of-freedom from the model are presented and the appropriateness of use is discussed. The strategies discussed herein are a momentum based approach and a velocity based approach. The momentum-based approach is derived from modeling discontinuous changes in model definition as instantaneous application (or removal) of constraints. The velocity-based approach is based on removing a degree-of-freedom when the associated generalized speed is zero. A Numerical example is included that demonstrates that both methods similarly characterize long-time conformational motion of a system.

3.1 Introduction

Numerical simulations of biopolymeric systems modeled using a fully atomistic approach are limited by their inherent size and complexity. With many important biological processes being relatively slow-acting, taking place on $O(10^{-3}) - O(10^{0})$ s and a typical time integration step on the order of femtoseconds, a large amount of computational resources are needed to simulate even simple systems. This coupled with the size of biomolecular problems, where systems contain 10^{3}–10^{7} or more

J. J. Laflin (✉) · K. S. Anderson · I. M. Khan
Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA
e-mail: laflij@rpi.edu

K. S. Anderson
e-mail: anderk5@rpi.edu

I. M. Khan
e-mail: khani4@rpi.edu

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_3,

atoms, makes fully atomistic simulations of complex systems impractical for scientists without access to premier computing facilities. For these reasons, techniques that facilitate molecular simulations of biopolymeric systems by improving the computational efficiency are a necessity to simulate systems that are large enough to be of interest for a sufficiently long time to gain insight into biological processes. Efficiency increases can be realized in two ways: (i) increasing the integration time step, and (ii) decreasing the amount of computational work done per time step. One method to increase temporal integration time-step size is to ignore (effectively freeze) high frequency intramolecular motions, since have been identified as not contributing significantly to the system's conformational motion. This technique can result in orders of magnitude increases in the time integration step.

Ignoring this motion is accomplished by removing the degrees-of-freedom associated with the intramolecular motion, provided their long-time relative average displacement is nearly zero. This is the essence of model reduction (coarsening). There are two main approaches to model coarsening: bead-based strategies and multibody-based strategies. In bead-based strategies, intramolecular degrees-of-freedom are effectively removed by replacing portions of the molecule with a spherical bead that represents the most essential physical properties of the atoms it replaces. This bead is then treated like a super-atom (particle) for the duration of the simulation. Early examples of this technique are demonstrated by Toxvaerd [20] and Padilla and Toxvaerd [15] and continued interest in coarse-grain molecular models is demonstrated by Praprotnik [18], Tozzini [21], and Chakrabarty [4]. Alternatively, the same degrees-of-freedom can be reduced using a multibody approach to model reduction [3, 5, 8, 14, 16, 17]. The inertia properties of the molecule are computed using the average atomic positions. The rotational equations of motion for the molecule are now included, but the motions permitted by these bodies relative to their adjacent neighbors is kinematically restricted by the nature of the bonds between them. Thus, the subsystem geometry, inertia properties, and associated forcing terms may be more accurately described by this articulated multibody model.

The motions associated with these articulated degrees-of-freedom are typically not high frequency in nature and are less than those in a reduced bead model. If these degrees-of-freedom are of a high frequency nature, this suggests that further model reduction should be performed. Including these rotational degrees-of-freedom has been shown to affect the translational acceleration of the associated regions and therefore possibly its overall conformational motion [17].

Figure 3.1 illustrates the topological differences resulting from the two approaches when creating a coarse-grained model of peptide kassinin. It is important to note that for this particular example both of these systems have the same number of degrees-of-freedom. However, for the bead-based coarsened model the rotational equations of motion are unimportant due to the particle nature of each super-atom. Therefore, the corresponding degrees-of-freedom are neglected.

A multibody coarse-graining strategy is easily implemented by fixing the average bond-length of stiff bonds. It is also clear that this would be appropriate for interactions where the bond strength is known to be large, such as carbon double-bonded to oxygen. However, there may be cases where the degrees-of-freedom associated

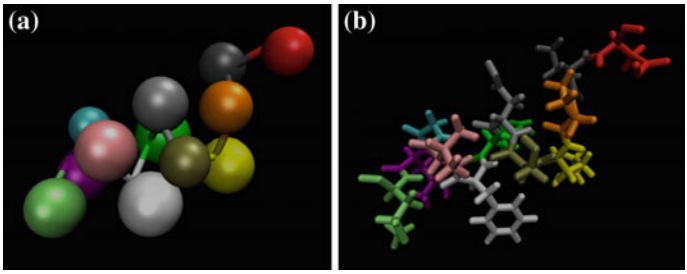


Fig. 3.1 Reduced model of kassinin. **a** Bead-based coarsening. **b** Multibody-based coarsening

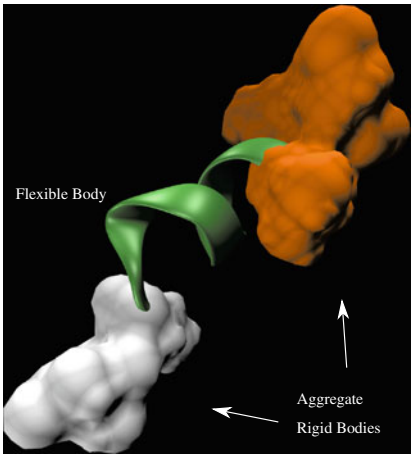


Fig. 3.2 Further model reduction of kassinin

with medium strength bonds can be removed with no effect on the overall motion of the system. Knowing when this is appropriate for low to medium strength bonds is difficult to predict prior to the simulation without considerable prior knowledge of the system's behavior. These types of bonds may exist between rigid bodies and removing the corresponding degrees-of-freedom could be used to further reduce the number of degrees-of-freedom in a system. Additionally, these bonds may be present between large numbers of atoms or small molecules where there is no clear rigid body. In this case, a flexible body may be created by removing those degrees-of-freedom. Figure 3.2 illustrates a further reduction in the number of degrees-of-freedom used to model Kassinin resulting in only three bodies, one flexible and two rigid.

Accurately predicting the behavior of many (frequently large) molecular systems does not require full atomistic resolution. This is because the overall conformational motion is generally independent of the high frequency intramolecular motion. In addition, as mentioned previously, it is not practical to perform an all atom simulation for these large systems. However, to study certain physical behavior it may be

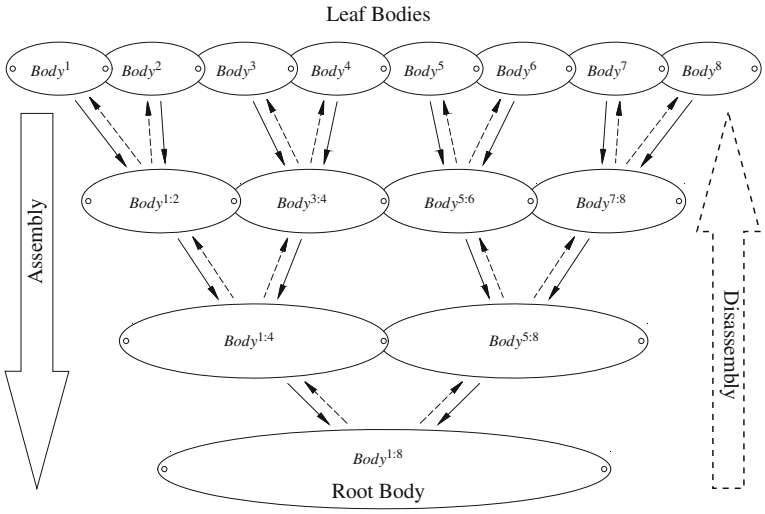


Fig. 3.3 DCA assembly and disassembly structure

desirable to maintain a high molecular, or even atomistic, resolution model in some regions of the model.

Due to the nature of large scale molecular simulations, particles must be able to be exchanged between the regions (computational domains) of different fidelity. This can create significant difficulty when trying to simulate some coarse-grained molecular systems. Praprotnik [18] demonstrates that this difficulty can be overcome by defining a criteria that specifies the resolution of the model instead of attempting to prescribe a particular resolution to various atoms or molecules. The simulation then automatically adjusts the model resolution in the appropriate regions based on this criteria. Poursina [17] demonstrates that reducing degrees-of-freedom in a static manner prior to the simulation, even for those degrees-of-freedom that are unimportant in a short simulation, can be significant in longer simulations and therefore can affect the overall conformational motion in a long-time simulation.

Adjusting the model resolution in various locations of the system as the simulation proceeds is easily accomplished when using a recursive Divide-and-Conquer (DCA) method to form and solve the equations of motion. The DCA was first applied to articulated multibody systems by Featherstone [6, 7]. This technique has been modified and extended in a number of ways, and has also been applied to other analyses associated with the design and simulation of multibody systems [1, 2, 10–12, 19]. A DCA-based method is well suited to accommodate adaptive changes in model fidelity because it recursively assembles and disassembles the inertia properties and resultant applied forcing terms of adjacent bodies connected by kinematic joints in a hierarchical manner, see Fig. 3.3. This makes the inertia properties and resultant forcing term parameterizations of any particular region or sub-domain of the problem readily available.

For multibody systems, changes in the number of degrees-of-freedom can be treated as instantaneous changes to the constraint definitions. With this method, constraint forces are impulsively applied or removed depending on the nature of the model fidelity transition. When performing model reduction with this method, a system wide impulse-momentum equation to determine the new generalized speeds and constraint forces must be performed. This is a straightforward, albeit computationally expensive, computation. Again, DCA methods can be used to reduce this computational expense to $O(n)$, where n is the number of bodies in the system, when applied serially or $O(\log(n))$ when applied in parallel [13]. Adding degrees-of-freedom to the system is not as simple, and will not be discussed in detail in this chapter. Again, a system wide impulse-momentum equation must be solved. However, in this case the solution is non-unique. Optimization can be used to select a solution from those which satisfy the impulse-momentum equations as shown by Khan et al. [9].

Alternatively, if the degree of freedom is removed when the generalized velocity is zero, then there is no need to solve a system-wide impulse-momentum equation. This technique avoids the previously mentioned $O(n)$ or $O(\log(n))$ (at best) computation. However, it requires that the reduction of a degree of freedom be postponed until the generalized speed associated with the degree of freedom is sufficiently close to zero. There can be cases in which postponing this model reduction is not computationally advantageous or generally appropriate, which will be discussed in further detail in subsequent sections. Therefore a strategy to perform model reduction should include both of the methods discussed.

3.2 Adaptive Model Reduction

Various criteria can be used to guide the reduction of degrees-of-freedom in areas of the model with motion occurring that is considered unimportant to the physical behavior being studied. These criteria can be derived from a variety of sources such as knowledge of the system or physical process being observed, physics-based metrics, chemical processes or reactions, and others. Maintaining a high model resolution at meaningful locations in a system, such as at a phase boundary or cellular membrane is an example of using knowledge of the system to create such a criterion. Statistical properties of the system parameters, such as the standard deviation of the relative joint displacement, may also guide model reduction. The selection of the criteria driving the model reduction may also influence the choice of method that is most appropriate to remove the degree or degrees-of-freedom.

3.2.1 Momentum-Based Model Reduction

As an example of when a momentum-based method would be desirable, consider the case mentioned previously, where the distance from a particular feature (e.g. cellular membrane) is the criteria for determining model reduction, and that it is proposed that the reduction of a degree of freedom should be made at the next

occurrence of its generalized speed reaching zero. In this case the reduction may be postponed an exceptionally long time or not occur, which may result in a large number of unnecessary degrees-of-freedom remaining in the simulation. Additionally, any situation in which it is desired to remove multiple degrees-of-freedom at a single instant would also require the use of the momentum-based approach.

Modeling changes in the number or location of degrees-of-freedom in a system as instantaneous application of constraints allows the degree-of-freedom to be removed at the next time step. However, it requires that an impulse-momentum balance be performed to determine the new constraint forces and the new generalized speeds of the post-transition system model. Since the impulse-momentum equation is the temporal integral of the equations of motion, the time-invariant inertia quantities (for rigid bodies) that are used to compute the solution to the equations of motion can also be used in the solution of these equations.

3.2.1.1 Rigid Bodies

Mukherjee and Anderson present a DCA-based method for modeling discontinuous changes in model definition of multibody systems [13]. Readers familiar with the use of the divide-and-conquer algorithm to form and solve the equations of motion will recognize the equations

$$\Delta v_1^k = \zeta_{11}^k \int_{t^-}^{t^+} F_{1c}^k \, dt + \zeta_{12}^k \int_{t^-}^{t^+} F_{2c}^k \, dt + \zeta_{13}^k \tag{3.1}$$

and

$$\Delta v_2^k = \zeta_{21}^k \int_{t^-}^{t^+} F_{1c}^k \, dt + \zeta_{22}^k \int_{t^-}^{t^+} F_{2c}^k \, dt + \zeta_{23}^k \tag{3.2}$$

as the impulse-momentum equations for two reference points (handles) on a generic rigid body ($body^k$). These reference points are typically located at the kinematic joints connecting the bodies. Those not familiar with a DCA-based approach are referred to the original work of Featherstone [6, 7], Mukherjee and Anderson who introduced a DCA-based method that uses the orthogonal complement of the allowable joint space to solve the equations of motion for closed-loop systems (O-DCA) [11], or Malczyk and Frączek who use the DCA with an augmented Lagrange multiplier approach [10].

In Eqs. (3.1) and (3.2) the quantities ζ_{ij}^k are the inverse inertial properties of body k at handle i (reference point i). These quantities are computed during the solution to the equations of motion (if a DCA-based strategy is being used to form and solve the equations of motion), and therefore can be re-used in these equations. Thus using a DCA-based method to solve both the equations of motion and the impulse-momentum equations increases the efficiency of the simulation. The change in the

spatial velocity of handle i due to the instantaneous imposition of constraint forces is represented by Δv_i^k.

The technique of Mukherjee and Anderson defines an allowable joint space motion map of joint j before $\left(P_{t_-}^j\right)$ and after $\left(P_{t_+}^j\right)$ the degree of freedom reduction. For example, consider removing the degree of freedom associated with a revolute joint between two bodies. In this case, the joint motion map before the reduction is

$$P_{t_-}^j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{3.3}$$

and the joint motion map after the reduction is

$$P_{t_+}^j = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{3.4}$$

The joint motion map for a completely locked joint should be a column vector of zeros to facilitate the matrix algebra of assembly and disassembly. The joint motion map contains the same number of unit spatial column vectors as there are degrees-of-freedom in the joint. Each of these unit vectors corresponds to a direction of motion allowed by the joint. Thus, in the example above where a single degree of freedom revolute joint is removed (locked) the joint motion map is changed from a 6×1 vector to the empty set. In general, the joint motion map is $6 \times \nu$, where ν is the number of degrees-of-freedom allowed by the joint.

The assembly and disassembly procedures integral to the divide-and-conquer algorithm compute the impulses and changes in the spatial velocities for all handles in the system. This is possible because the terminal connections of the system are known and the joint motion maps for all joints are known. Computing the instantaneous changes in the generalized speeds (u) is easily done using the kinematic relation between the spatial velocities at the handles and the generalized speeds. The kinematic relation between the spatial velocity and the generalized speed is

$$v_1^{k+1} - v_2^k = P^j u^j. \tag{3.5}$$

The difference in the spatial velocities of the reference points coincident with the joint connecting adjacent bodies k and $k+1$ is the generalized speed projected onto the joint motion map. Using a similar relation, the change in the generalized speed is computed from the change in the spatial velocity by

$$\Delta v_1^{k+1} - \Delta v_2^k = P_{t_+}^j u_{t_+}^j - P_{t_-}^j u_{t_-}^j \tag{3.6}$$

for joints in which there are changes to the number of degrees-of-freedom, and

$$\Delta v_1^{k+1} - \Delta v_2^k = P^j \left(u_{t_+}^j - u_{t_-}^j\right) \tag{3.7}$$

for joints that are unmodified. It should be noted that after the assembly and disassembly processes have been completed the only unknown quantity in Eqs. (3.6) and

(3.7) are the generalized speeds after the model transition $\left(u^j_{t_+}\right)$. Finally, the change in generalized speed can be computed by

$$u^j_{t_+} = \left(P^j_{t_+}\right)^T \left(\Delta v_1^{k+1} - \Delta v_2^k + P^j_{t_-} u^j_{t_-}\right) \tag{3.8}$$

or

$$u^j_{t_+} = \left(P^j\right)^T \left(\Delta v_1^{k+1} - \Delta v_2^k\right) + u^j_{t_-}. \tag{3.9}$$

3.2.1.2 Flexible Bodies

Khan et al. [9] details the process of removing or adding degrees-of-freedom to flexible bodies in the DCA-framework. Modeling the change in the degrees-of-freedom of a flexible body is performed in a similar manner, i.e. the equations of motion for two reference points on a body are written in terms of the inertial properties and the instantaneous applied constraints. The impulse-momentum equation for two handles on a flexible body written using a floating frame of reference approach are

$$\Gamma^f V_1^f - \Gamma^c V_1^c = \begin{bmatrix}\gamma_R \\ \gamma_F\end{bmatrix}_1^c \int_{t_-}^{t_+} F^k_{1c}\, dt + \begin{bmatrix}\gamma_R \\ \gamma_F\end{bmatrix}_2^c \int_{t_-}^{t_+} F^k_{2c}\, dt \tag{3.10}$$

and

$$\Gamma^f V_2^f - \Gamma^c V_2^c = \begin{bmatrix}\gamma_R \\ \gamma_F\end{bmatrix}_1^c \int_{t_-}^{t_+} F^k_{1c}\, dt + \begin{bmatrix}\gamma_R \\ \gamma_F\end{bmatrix}_2^c \int_{t_-}^{t_+} F^k_{2c}\, dt. \tag{3.11}$$

This method may use modal coordinates and admissible shape functions or element displacement functions, as done in Finite Element Analysis, to model the flexibility of the body coupled with the rigid body motion. In Eqs. (3.10) and (3.11) the Γ terms are the mass and inertial matrices. V_i contains the spatial velocity associated with reference point i (v_i) as well as the generalized speeds associated with the modal degrees-of-freedom ($\dot{q}$),

$$V_i = \begin{bmatrix} v_i \\ \dot{q} \end{bmatrix}. \tag{3.12}$$

All terms with the superscript c are associated with the coarse model (post-reduction) and the terms with the superscript f are associated with the fine model (pre-reduction).

The change in generalized speeds associated with the degrees-of-freedom of the joint can be computed from Eqs. (3.8) and (3.9). The new mode speeds can be computed by partitioning the matrix equations Eqs. (3.10) and (3.11). To facilitate this, the inertial matrix Γ is partitioned into its rigid components (Γ_{RR}), flexible compo-

nents (Γ_{FF}), and the terms that couple the rigid and flexible motion (Γ_{RF} and Γ_{FR}). By doing so, the new mode speeds after model reduction can then be computed as

$$\dot{q}^c = \left(\Gamma^c_{FF}\right)^{-1} \left[\gamma^c_{F1} \int_{t-}^{t+} F_{1c}\, dt + \gamma^c_{F2} \int_{t-}^{t+} F_{2c}\, dt + \Gamma^f_{FR} v^f_1 + \Gamma^f_{FF} \dot{q}^f - \Gamma^c_{FR} v^c_1 \right]. \tag{3.13}$$

The derivation of these terms is detailed by Mukherjee and Anderson [12].

3.2.2 Zero-Velocity Model Reduction

An alternate strategy for model coarsening in biomolecular systems makes use of the periodic nature of motion about a varying mean, which is generally present in biopolymer systems. This strategy simply performs the model transition once the internal metric has indicated that elimination of the degree of freedom is appropriate. This strategy proposes that at the target joint the relative motion is locally oscillatory in nature. Additionally, this method assumes that after a degree of freedom associated with a particular joint or modal coordinate has been targeted to be ignored (removed), the model transition can be postponed a short time.

3.2.2.1 Rigid Bodies

With the pre-transition $\left(u^j_{t+}\right)$ and post-transition $\left(u^j_{t+}\right)$ generalized speeds both zero, Eqs. (3.6) and (3.7) both become

$$\Delta v^{k+1}_1 - \Delta v^k_2 = 0. \tag{3.14}$$

This equation is satisfied if the change in spatial velocity at coincident reference points is zero or the change in spatial velocity of these reference points are equal and opposite. However, it is known that the latter scenario is not the case. Therefore the impulse-momentum equations for the reference points becomes

$$0 = \zeta^k_{11} \int_{t^-}^{t^+} F^k_{1c}\, dt + \zeta^k_{12} \int_{t^-}^{t^+} F^k_{2c}\, dt + \zeta^k_{13} \tag{3.15}$$

and

$$0 = \zeta^k_{21} \int_{t^-}^{t^+} F^k_{1c}\, dt + \zeta^k_{22} \int_{t^-}^{t^+} F^k_{2c}\, dt + \zeta^k_{23}. \tag{3.16}$$

Since there are no applied external impulses, $\zeta_{i3} = 0$. Also it is known that ζ_{i1} and ζ_{i2} are both non-zero. Therefore, the impulses at the handles must be zero.

3.2.2.2 Flexible Bodies

Removing the degree of freedom associated with a kinematic joint between flexible bodies at zero velocity follows similarly. This can be seen by writing the matrix equation Eq. (3.10) separately as the two submatrix equations

$$\overbrace{\left(\Gamma^c_{RR} v^c_1 - \Gamma^f_{RR} v^f_1\right)}^{0} + \overbrace{\left(\Gamma^c_{RF} \dot{q}^c - \Gamma^f_{RF} \dot{q}^f\right)}^{0} = \gamma^c_{R1} \int_{t-}^{t+} F_{1c}\, dt + \gamma^c_{R2} \int_{t-}^{t+} F_{2c}\, dt \tag{3.17}$$

and

$$\overbrace{\left(\Gamma^c_{RF} v^c_1 - \Gamma^f_{RF} v^f_1\right)}^{0} + \overbrace{\left(\Gamma^c_{FF} \dot{q}^c - \Gamma^f_{FF} \dot{q}^f\right)}^{0} = \gamma^c_{F1} \int_{t-}^{t+} F_{1c} dt + \gamma^c_{F2} \int_{t-}^{t+} F_{2c}\, dt. \tag{3.18}$$

This is possible because it is known from the rigid body discussion in Sect. 3.2.1.1 that removing the generalized coordinate associated with the motion of a kinematic joint at zero generalized speed results in no change of the spatial velocity at the reference points on the body. Additionally, if there is no change in the spatial velocities the impulses at the reference points must again be zero because γ_R and γ_F are not zero. Therefore, there will not be any change in the modal speeds. Similarly, by removing a degree of freedom associated with the modal coordinates at zero modal speed, there is no effect on the spatial velocities.

3.3 Numerical Example

A simple rigid body system, shown in Fig. 3.4, is used to investigate the differences between the momentum-based and velocity-based methods for performing model reduction when the joint relative motion drops below a certain threshold, for a given window in time. In this problem, a force equivalent to $0.05\, g$ is applied in the $-\hat{n}_2$ direction to create the oscillatory aspects of the local motion, where $g = 9.81$ N/m^2. The joints are locked when the standard deviation of the generalized coordinate drops below 1×10^{-4} for the previous 100 time steps. Model reduction is restricted to after 1 s of time has elapsed. The joints that meet this locking criteria are locked with both the zero-velocity method and the instantaneous momentum-based method. A custom fixed time-step fourth order Runge-Kutta integrator was used. This motion

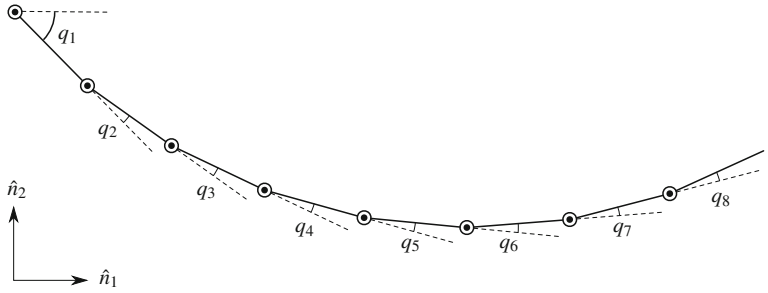


Fig. 3.4 Test problem

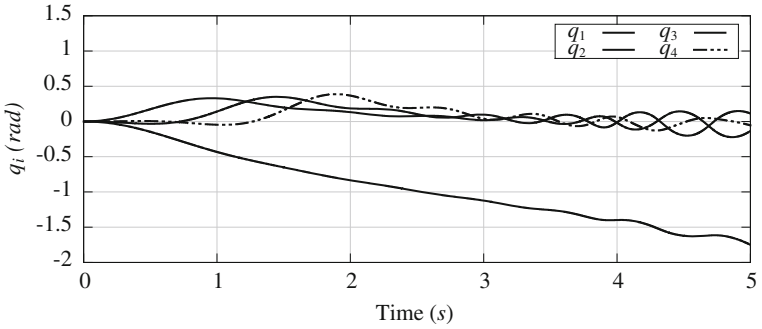


Fig. 3.5 Without model reduction

simulates one end of a chain-type biopolymer docked at one end and oscillating due to the far field forces.

Figure 3.5 shows the motion of the generalized coordinates throughout the course of the simulation. The generalized coordinates that are locked by either method are removed from the plot so that the long term motion of the remaining degrees-of-freedom can be seen in the case where model reduction is not used. This figure shows expected behavior of all joints oscillating around zero, the exception is that the first joint will oscillate at a much lower time scale, so low that it is not seen in the time scale of the figure.

Figure 3.6 shows the same degrees-of-freedom over time. However, this figure shows the cases where the zero-velocity (Fig. 3.6a), and the impulse-momentum (Fig. 3.6b) methods are used for model reduction. The joints that are locked, and the times at which they are locked are reported in the respective figure. The locked degrees-of-freedom are represented by solid lines and the time at which they are locked is marked with the asterisk, both of which are not reported in the figure key. The difference in times at which the joints are locked in Fig. 3.6a, b is due to the zero-velocity method of model reduction waiting for the joint velocity to reach near zero before it is removed from the model. In this numerical example, once a joint has been identified to be locked, the degree of freedom is removed at the next time step

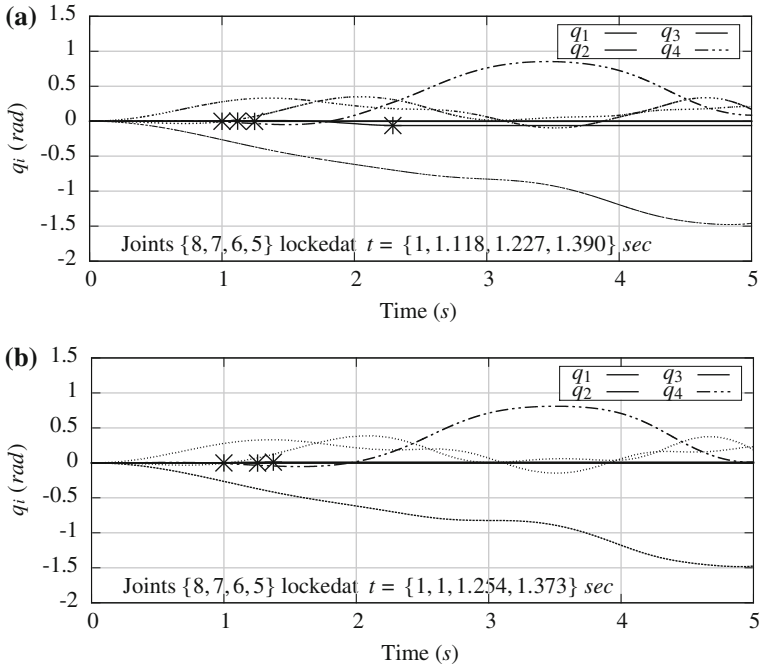


Fig. 3.6 Zero velocity model reduction

that it has near zero velocity. The model reduction criteria are not re-evaluated. For a more robust implementation, if the joint is not locked within a specified window of time perhaps the locking criteria should be re-evaluated.

Comparing Fig. 3.6a, b it is clear that both methods result in similar overall long-term motion. Direct comparison of Fig. 3.5 with Fig. 3.6a or b would be inappropriate since no method of adding fidelity (unlocking a joint) was included in this example, therefore this is not a truly adaptive example. Figure 3.5 was included to demonstrate that the unmodified behavior of the system is aligned with its expected behavior.

3.4 Discussion

The advantage of a zero-velocity approach is that there is no impulse-momentum balance to perform, which will provide a significant computational savings. Furthermore, the energy of the system will not decrease with each model reduction performed with this approach. The energy of the ignored degrees-of-freedom will be stored in the potential energy of the system. This will eliminate the "cooling effect" of many model reductions executed throughout the course of the simulation. This

effect is the kinetic energy losses manifested by temperature reduction, which is an important system property in molecular systems.

The zero-velocity method is particularly advantageous if the reverse transition is performed and the fidelity of the model is increased by adding this degree of freedom back into the model. In this situation, the frozen degree of freedom is simply released and the stored potential energy of this mode of motion is permitted to act on the system. This saves the considerable computational expense associated with the optimization problem that must be performed to determine a suitable solution to the impulse-momentum balance of the many possible solutions. This process is explained in more detail by Khan et al. [9].

In terms of computational cost, this technique becomes particularly advantageous when the number of degrees-of-freedom that would be reduced is significantly less than the number of degrees-of-freedom in the entire system. For example, if there are 10^3 degrees-of-freedom in the system and it is desired to remove 1 of them, immediately reducing the joint at non-zero velocity requires an $O(10^3)$ computation to perform the system-wide impulse-momentum balance. By postponing this model reduction until the joint velocity is near zero, no momentum-balance needs to be performed, therefore there is no cost of reducing this joint.

This issue is particularly relevant when implementing an adaptive model reduction method. The goal of using adaptive changes in model resolution is that the computing resources are used where most appropriate. Therefore, model resolution in regions which are of little interest is reduced. Adaptive model reduction may use a variety of metrics to determine if a joint should be locked or unlocked. An intuitive method for biomolecular systems may be to monitor the standard deviation of the relative joint angle over a given sliding window of time [17]. If a momentum-based method is naively implemented, and the number of degrees-of-freedom reduced from the model is not significant enough to result in computational savings over the course of the simulation, the computational cost of the simulation could increase dramatically, approaching $O(n^2)$ in the worst case, when an $O(n)$ method is used to compute the impulse-momentum balance and the equations of motion. This places practical limits on the minimum number of degrees-of-freedom that can be reduced, so that model reduction does not actually add to the cost of the simulation.

Conversely, if a large number of degrees-of-freedom are targeted for reduction the most efficient method of model reduction might be the impulse-momentum balance method which reduces all of these degrees-of-freedom at the next time step. This depends on the frequency of the periodic nature of the joint relative motion. If the frequency is relatively low, a significant number of time steps may pass before the joint velocity falls below the threshold set and is reduced, if at all. If that takes place for relatively large number of degrees-of-freedom, e.g. $O(n)$, $O(n) + \delta O(n)$ extra computations may have taken place before all of these joints are reduced.

Another example in which a momentum-based method must be used is when transitioning from a certain system subdomain model to a fundamentally different model type. This could be encountered in a transition from a set of rigid bodies to a single flexible body, then a momentum-based approach must be used. This is because this type of transition between such different subdomain model types requires that

multiple independent degrees-of-freedom be removed simultaneously. As a consequence, not all generalized speed associated with these degrees-of-freedom will be simultaneously zero.

The momentum-based method avoids a potential disadvantage of the zero-velocity method. This disadvantage is that the degree of freedom associated with kinematic joints and deformation of flexible bodies is typically at zero generalized speed when the associated degree of freedom is at maximum deflection or relative motion. This may cause unrealistic geometric configurations if care is not used in selection of the criteria used to ignore a degree of freedom.

This suggests a model reduction strategy that uses both methods where appropriate is needed. A method of performing impulse-momentum balance, such as [13], would already be part of the computational machinery to deal with collisions. Therefore, with little extra effort, a zero-velocity method could be incorporated to provide a more robust handling of model reduction.

Both methods of model reduction produce comparable results when examining the long-time macro-scale motion of the system. The notable difference in these methods is the computational cost. In the above example, 4 degrees-of-freedom are removed, using the impulse-momentum method this would add $O(3n)$ (because joints 7 and 8 are locked at the same time step) to the computational cost of the simulation. The method of performing model transition at zero velocity reduces the same degrees-of-freedom from the model, although at different times, at nearly zero additional computational cost.

This demonstrates the principle difficulty with adaptive impulse-momentum based model reduction for biopolymers. If done wisely, this method can realize significant computational savings over the course of a simulation. As an example, consider a system where the number of degrees-of-freedom are reduced from 200 to 20 at a single time step. In such a situation, the impulse-momentum based methods reduce 180 degrees-of-freedom for $O(n)$ computational cost. If this is done early enough in the simulation, there would be significant overall computational savings.

Alternatively if these degrees-of-freedom are reduced intermittently throughout the simulation, there would be $O(180n)$ additional computational cost in the worst case. This would result in an overall $O(n^2)$ computational cost. In this situation, the zero-velocity method may provide significant computational savings. This is however subject to the frequency of the motion at the joint that is targeted for reduction, as significant time may elapse before the joint velocity becomes zero.

3.5 Conclusion

An adaptive method of performing model reduction should include both the zero-velocity method and the momentum-based method, since it would generally not be known a priori when reductions in the number of degrees-of-freedom would occur. Some metric of the time remaining in the simulation and number of degrees-of-freedom that are targeted to be removed could be used to guide the decision of which

method to use. Using both methods of model reduction should not involve a great deal of effort since a method that solves the impulse-momentum equations must exist to handle collisions, and incorporating the zero-velocity method can be done with minimal programming effort. The way in which these two methods are used may depend heavily on the metric that is used to guide model reduction.

References

1. Bhalerao KD, Crean C, Anderson KS (2011) Hybrid complementarity formulations for robotics applications. ZAMM J Appl Math Mech (Zeitschrift für Angewandte Mathematik und Mechanik) 91(5):386–399
2. Bhalerao KD, Poursina M, Anderson KS (2009) An efficient direct differentiation approach for sensitivity analysis of flexible multibody systems. Multibody Syst Dyn 23(2):121–140
3. Carnevali P, Toth G, Toubassi G, Meshkat SN (2003) Fast protein structure prediction using Monte Carlo simulations with modal moves. J Am Chem Soc 125:14244–14245
4. Chakrabarty A, Cagin T (2010) Coarse grain modeling of polyimide copolymers. Polymer 51(12):2786–2794
5. Chun HM, Padilla CE, Chin DN, Watenabe M, Karlov VI, Alper HE, Soosaar K, Blair KB, Becker OM, Caves LSD, Nagle R, Haney DN, Farmer BL (2000) MBO(N)D: a multibody method for long-time molecular dynamics simulations. J Comput Chem 21(3):159–184
6. Featherstone R (1999) A divide-and-conquer articulated-body algorithm for parallel O(log(n)) calculation of rigid-body dynamics. Part 1: basic algorithm. Int J Rob Res 18(9):867–875
7. Featherstone R (1999) A divide-and-conquer articulated-body algorithm for parallel O(log(n)) calculation of rigid-body dynamics. Part 2: trees, loops, and accuracy. Int J Rob Res 18(9):876–892
8. Jain A, Vaidehi N, Rodriguez G (1993) A fast recursive algorithm for molecular dynamics simulation. J Comput Phys 106(2):258–268
9. Khan IM, Anderson KS (2013) Khan IM, Poursina M, Anderson KS (2013) Model transitions and optimization problem in multiflexible-body systems: Application to modeling molecular systems. Comput Phys Commun 184(7), 1717–1728
10. Malczyk P, Frączek J (2012) A divide and conquer algorithm for constrained multibody system dynamics based on augmented Lagrangian method with projections-based error correction. Nonlinear Dyn 70(1):871–889
11. Mukherjee RM, Anderson KS (2006) Orthogonal complement based divide-and-conquer algorithm for constrained multibody systems. Nonlinear Dyn 48(1–2):199–215
12. Mukherjee RM, Anderson KS (2007) A logarithmic complexity divide-and-conquer algorithm for multi-flexible articulated body dynamics. J Comput Nonlinear Dyn 2(1):10
13. Mukherjee RM, Anderson KS (2007) Efficient methodology for multibody simulations with discontinuous changes in system definition. Multibody Syst Dyn 18(2):145–168
14. Mukherjee RM, Crozier PS, Plimpton SJ, Anderson KS (2008) Substructured molecular dynamics using multibody dynamics algorithms. Int J Nonlinear Mech Nonlinear Mech Dyn Macromol 43(10):1040–1055
15. Padilla P, Toxvaerd S (1991) Structure and dynamical behavior of fluid n-alkanes. J Chem Phys 95(1):509
16. Poursina M (2011) Robust framework for the adaptive multiscale modeling of biopolymers, Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY
17. Poursina M, Bhalerao KD, Flores SC, Anderson KS, Laederach A (2011) Strategies for articulated multibody-based adaptive coarse grain simulation of RNA. Methods Enzymol 487:73–98
18. Praprotnik M, Delle Site L, Kremer K (2005) Adaptive resolution molecular-dynamics simulation: changing the degrees of freedom on the fly. J Chem Phys 123(22):224106

19. Redon S, Galoppo N, Lin MC (2005) Adaptive dynamics of articulated bodies. ACM Trans Graph 24(3):936–945
20. Toxvaerd S (1989) Molecular dynamics calculation of the equation of state of liquid propane. J Chem Phys 91(6):3716
21. Tozzini V (2005) Coarse-grained models for proteins. Curr Opin Struct Biol 15(2):144–150

Chapter 4
A Mortar Method Combined with an Augmented Lagrangian Approach for Treatment of Mechanical Contact Problems

Federico J. Cavalieri, Olivier Brüls and Alberto Cardona

Abstract This work presents a mixed penalty-duality formulation from an augmented Lagrangian approach for treating the contact inequality constraints. The augmented Lagrangian approach allows to regularize the non differentiable contact terms and gives a C^1 differentiable saddle-point functional. The relative displacement of two contacting bodies is described in the framework of the Finite Element Method (FEM) using the mortar method, which gives a smooth representation of the contact forces across the bodies interface. To study the robustness and performance of the proposed algorithm, validation numerical examples with finite deformations and large slip motion are presented.

4.1 Introduction

Contact mechanics is present in a wide range of mechanical engineering applications, and numerous works have dealt with the numerical solution of contact problems. New advances and techniques, including friction, large displacements, plasticity, and wear, are constantly being introduced. However, there is not yet a completely robust contact algorithm suitable for a wide range of applications in contact mechanics.

F. J. Cavalieri · A. Cardona (✉)
Centro de Investigación de Métodos Computacionales (CIMEC), Universidad Nacional del Litoral—CONICET, Predio Conicet-Santa Fe, Colectora Ruta Nac 168 s/n/ Paraje El Pozo, 3000 Santa Fe, Argentina
e-mail: acardona@conicet-santafe.gov.ar

F. J. Cavalieri
e-mail: fcavalieri@conicet-santafe.gov.ar

O. Brüls
Department of Aerospace and Mechanical Engineering (LTAS), University of Liège, Chemin des Chevreuils, 1 (B52), 4000 Liège, Belgium
e-mail: o.bruls@ulg.ac.be

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_4,

Mathematically, the contact problem can be interpreted as a physical system defined by a variational inequality. The solution of frictionless or frictional contact problems corresponds to minimize the total energy of the system subjected to inequalities constraints associated to the normal and tangential components of the traction and distance vectors, respectively.

In the framework of the FEM, the node-to-segment technique is widely used in many commercial finite element codes for the description of the relative displacement between two contacting bodies. In this case, a node of one body (the slave) is associated with a segment or a surface of another body (the master). An extensive list of references with different variants and practical applications for this method can be found in the books of Wriggers [46] or Laursen [29].

The main drawback of the node-to-segment approach is that it is not able to transmit a constant stress field from one body to the other when the meshes are non-conforming, i.e. it does not pass the contact patch test; therefore, it introduces errors in the solution independently of the mesh discretization of the contacting bodies [33]. Furthermore, when the slave nodes slide from one to another master segment, the solution shows jumps in the contact stress field due to the enforcement of the discrete contact constraints. The double pass node-to-segment approach satisfies the contact patch test, but it can "lock" due to the over-restriction introduced in the formulation [38].

Other method introduced to simulate contact mechanics problems is the so-called segment-to-segment approach, where the segment of one body (the master segment) is associated with a segment of the other body (the slave segment). Most of the segment-to-segment methods use some kind of intermediate surface or projection surface. The surface-to-surface mortar strategy, was originally proposed as a domain decomposition method and was used to solve finite element problems with non conforming discretizations. An important characteristic of the mortar method is that it is only one pass and verifies the contact patch tests. Therefore, a double pass proposal would be completely unnecessary. Specifically, the first work that has used the mortar method was published by Bernardi et al. [4], where the authors demonstrated the stability properties related with the Babuska-Brezzi conditions [5]. The first proposals of the mortar method used in engineering applications were introduced in the frame of small deformations [30]. Then, other authors, such as Puso and Laursen [37, 38], Hüeber and Wohlmuth [19], Hüeber et al. [25], Fischer and Wriggers [14], Hartmann and Ramm [18], Popp et al. [34, 35], Hesch and Betsch [24], Hüeber and Wohlmuth [20], Cichosz and Bischoff [11], Hesch and Betsch [23], Cavalieri et al. [9], have extended the mortar approximation to problems with finite deformations, large sliding or time dependent problems. The works of Papadopoulos and Solberg [32], Jones and Papadopoulos [26], Solberg and Papadopoulos [41] and Solberg et al. [40] are not known as mortar methods; however, they incorporate all the characteristics to be classificated as mortar [42]. Recently, Temizer [42], De Lorenzis et al. [13], and Kim and Youn [28] applied the mortar methods using isogeometric analysis.

The treatment of the restrictions in contact problems could be addressed with different strategies: by using a penalty approach, Lagrange multipliers, augmented Lagrangian techniques, and others. Several works have proposed friction contact

algorithms by using the mortar method. For example, Puso and Laursen [39] proposed a mortar method with a penalty regularization within an augmented Lagrangian scheme [27]. The works of Yang et al. [47], and Fischer and Wriggers [15] present a penalty method for the regularization of the variational problem and a mortar approach to describe the contact kinematics. Heintz and Hansbo [22] proposed a stabilized Lagrange multiplier method based on a global polynomial multiplier for the finite element solution of non-linear elastic contact problems with non-matching grids in 2D. More recently, Gitterle et al. [16] proposed a two-dimensional finite deformation frictional contact formulation based on a mortar formulation. In this case, the enforcement of contact constraints is reached with dual Lagrange multipliers. By using the so-called dual mortar method, the Lagrange multipliers can be eliminated by static condensation from the set of linear algebraic equations [45]. However, the dual method may lack robustness, e.g., when contact surfaces have large curvatures or when one contacting body slips over an edge of the other contacting body. A more robust version of this method has been presented recently by Popp et al. [36].

De Lorenzis et al. [12] focused on the application of NURBS-based isogeometric analysis to Coulomb frictional contact problems between deformable bodies in the context of large deformations and using the classical return mapping algorithm. A three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS in the finite deformation regime is presented by Temizer et al. [43, 44]. In both works, a penalty approach supplemented by Uzawa augmentations is implemented for the regularization of the contact constraints.

In this work, unlike the above mentioned proposals, the augmented Lagrangian technique, introduced by Alart and Curnier [1], is combined with a mortar approach to solve three dimensional frictionless and frictional contact problems. The form of the augmented Lagrangian approach allows the regularization of non-differentiable contact terms. The resulting equations involving the Lagrange multipliers are linearized and solved using a standard *Newton-Raphson*-like method. The mortar method leads to a correct representation of the stress fields across the contact interface.

To validate the proposed method, firstly a numerical example is presented to demonstrate that the algorithm satisfies the contact patch test. Then, another example including friction is proposed to validate the algorithm with a classical frictional benchmark problem. Finally, the algorithm is used in the regime of large displacements to demonstrate its applicability and robustness in more general applications. It is remarkable that in all cases the normal and tangential stress solutions are smooth, even in non-conforming meshes.

4.2 Problem Description

Figure 4.1 depicts the contact problem. The contacting bodies $\mathscr{B}^\alpha$, $\alpha = 1, 2$, occupy the open set $\Omega^\alpha \subset \mathbb{R}^3$ in the reference configuration and $\Omega_a^\alpha \subset \mathbb{R}^3$ in the current configuration. In the reference configuration, the external surface $\partial\Omega^\alpha$ is divided

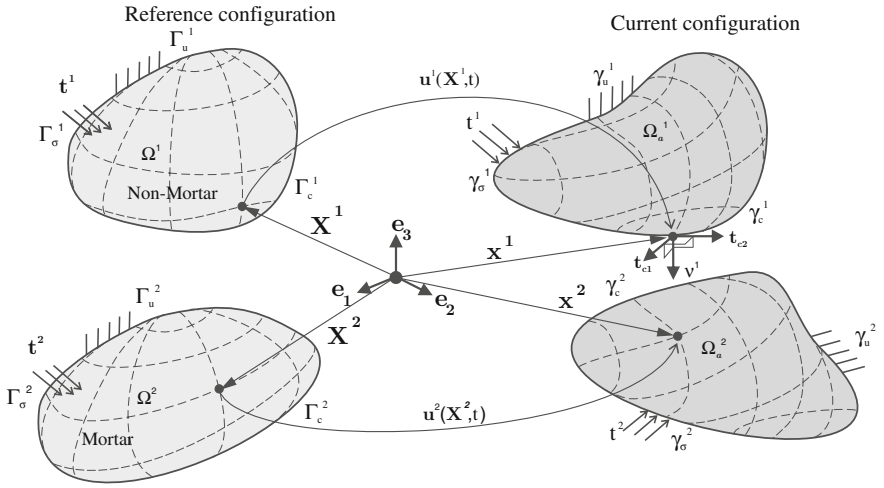


Fig. 4.1 Two-body contact problem in the framework of large deformation

into three disjoint boundaries: Γ_u^α where the body is fixed, Γ_σ^α where the surface traction vector is acting, and Γ_c^α which represents the contact boundary. The same boundaries in the current configuration are denoted γ_u^α, γ_σ^α and γ_c^α, respectively.

In the reference configuration, the material points for each solid are denoted by the position vector $\boldsymbol{X}^\alpha \in \Omega^\alpha$, while in the current configuration they are given by the vector $\boldsymbol{x}^\alpha \in \Omega_a^\alpha$. Then, the movement of both bodies is described by the displacement field $\boldsymbol{u}^\alpha$ which is related to the reference and current positions by $\boldsymbol{x}^\alpha = \boldsymbol{X}^\alpha + \boldsymbol{u}^\alpha$. The total potential energy for the contacting bodies $\mathscr{B}^\alpha$, is given by

$$\Pi = \Pi_{\text{cont}} + \Pi_{\text{int,ext}}, \tag{4.1}$$

where Π_{cont} is the contact potential energy, and $\Pi_{\text{int,ext}}$ represents the potential energy of the external and internal loads. In the framework of finite deformations, $\Pi_{\text{int,ext}}$ yields

$$\Pi_{\text{int,ext}} = \sum_{\alpha=1}^{2} \left[\int_{\Omega^\alpha} \left(\boldsymbol{E}^\alpha : \boldsymbol{S}^\alpha - \boldsymbol{b}^\alpha \cdot \boldsymbol{u}^\alpha \right) d\Omega - \int_{\Gamma_N^\alpha} \hat{\boldsymbol{t}}^\alpha \cdot \boldsymbol{u}^\alpha \, d\Gamma \right]. \tag{4.2}$$

Here, $\boldsymbol{b}^\alpha$ is the body force vector in Ω^α, $\hat{\boldsymbol{t}}^\alpha$ is the prescribed traction vector, $\boldsymbol{S}^\alpha$ is the second Piola-Kirchhoff stress tensor, which is related to the deformations by an appropriate constitutive law, for example,

$$\boldsymbol{S}^\alpha = \boldsymbol{C}^\alpha : \boldsymbol{E}^\alpha, \tag{4.3}$$

where $\boldsymbol{C}$ is the fourth-order constitutive stress tensor, and $\boldsymbol{E}$ is the Green-Lagrange deformation tensor defined as

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right), \tag{4.4}$$

with $\boldsymbol{F}$ representing the deformation gradient. This work will put emphasis on the solution of contact equations; then, no details about the modeling of solid deformation will be presented. The potential energy due exclusively to the contact is given by

$$\Pi_{\text{cont}} = -\sum_{\alpha=1}^{2}\int_{\gamma_c^\alpha} \bar{\boldsymbol{t}_c}^{\alpha}\cdot\boldsymbol{x}^\alpha d\gamma, \tag{4.5}$$

where $\bar{\boldsymbol{t}_c}^{\alpha}$ is the Cauchy traction vector of the body $\mathcal{B}^\alpha$ in the current configuration. Assuming lineal momentum balance at the contact surface, $\bar{\boldsymbol{t}_c}^{1}d\gamma^1 = -\bar{\boldsymbol{t}_c}^{2}d\gamma^2$ the contact potential energy can be simplified as

$$\Pi_{\text{cont}} = -\int_{\gamma_c^1} \bar{\boldsymbol{t}_c}^{\alpha}\cdot\left(\boldsymbol{x}^1 - \boldsymbol{x}^2\right)d\gamma, \tag{4.6}$$

Instead of using the Cauchy traction vector $\bar{\boldsymbol{t}_c}^{1}$, a Lagrange multiplier $\boldsymbol{\lambda}$ is introduced such that $\boldsymbol{\lambda} = -\bar{\boldsymbol{t}_c}^{1}$. Then, the contact potential energy in Eq. (4.6) is re-written as

$$\Pi_{\text{cont}} = \int_{\gamma_c^1} \boldsymbol{\lambda}\cdot\left(\boldsymbol{x}^1 - \boldsymbol{x}^2\right)d\gamma. \tag{4.7}$$

The finite element method is used to discretize the body domains. The contact surface of each body and the traction vector are parameterized as follows [37],

$$\boldsymbol{x}^\alpha = \sum_{A=1}^{n^\alpha} N_A^\alpha(\boldsymbol{\xi}^\alpha)\boldsymbol{x}_A^\alpha, \quad \alpha = 1, 2, \qquad \boldsymbol{\lambda} = \sum_{A=1}^{n^1} N_A^1(\boldsymbol{\xi}^1)\boldsymbol{\lambda}_A, \tag{4.8}$$

where $\boldsymbol{x}_A^\alpha \in \gamma_c^\alpha \to \mathbb{R}^3$ are the nodal coordinates, $\boldsymbol{\xi}^\alpha$ are the coordinates of the Gauss points, n^α is the number of nodes in γ_c^α, and $N_A^\alpha : \gamma_c^\alpha \to \mathbb{R}$ are the classical shape functions used in the FEM discretization. As usually mentioned in the literature, γ_c^1 and γ_c^2 are the non-mortar and mortar surfaces in the current configuration, respectively. The Lagrange multiplier $\boldsymbol{\lambda}_A \in \gamma_c^1 \to \mathbb{R}^3$ is discretized with the same shape functions used to approximate the geometry and the displacements. By using Eqs. (4.8) and (4.7), the contact potential energy in the framework of FEM is expressed as

$$\Pi_{\text{cont}} = \sum_{A=1}^{n^1} \boldsymbol{\lambda}_A \cdot \left(\sum_{B=1}^{n^1} \int_{\gamma_c^1} N_A^1(\boldsymbol{\xi}^1) N_B^1(\boldsymbol{\xi}^1)\, d\gamma\, \boldsymbol{x}_B^1 - \sum_{C=1}^{n^2} \int_{\gamma_c^1} N_A^1(\boldsymbol{\xi}^1) N_C^2(\boldsymbol{\xi}^2)\, d\gamma\, \boldsymbol{x}_C^2 \right), \tag{4.9}$$

where the term in parenthesis in Eq. (4.9), can be interpreted as a measure of the interpenetration or average gap corresponding to the node A. Therefore,

$$\Pi_{\text{cont}} = \sum_{A=1}^{n^1} \boldsymbol{\lambda}_A \cdot g_A, \tag{4.10}$$

with

$$g_A = \left(\sum_{B=1}^{n^1} \int_{\gamma_c^1} n_{AB}^1\, d\gamma\, \boldsymbol{x}_B^1 - \sum_{C=1}^{n^2} \int_{\gamma_c^1} n_{AB}^2\, d\gamma\, \boldsymbol{x}_C^2 \right), \tag{4.11}$$

namely the gap vector at node A. Here, n_{AB}^1 and n_{AB}^2 are the weight factors defined as

$$n_{AB}^1 = \int_{\gamma_c^1} N_A^1(\boldsymbol{\xi}^1) N_B^1(\boldsymbol{\xi}^1)\, d\gamma, \qquad n_{AC}^2 = \int_{\gamma_c^1} N_A^1(\boldsymbol{\xi}^1) N_C^2(\boldsymbol{\xi}^2)\, d\gamma, \tag{4.12}$$

and computed by an assembly algorithm as presented in [8, 37].

4.3 Augmented Lagrangian Formulation for Frictionless Contact Problems

Assuming frictionless contact problems, only the normal direction of the traction vector $\boldsymbol{\lambda}$ and the gap g at node A are considered:

$$\lambda_{NA} = \boldsymbol{\nu}_A \cdot \boldsymbol{\lambda}_A, \qquad g_{NA} = \boldsymbol{\nu}_A \cdot g_A, \tag{4.13}$$

with $\boldsymbol{\nu}_A$ being the inward normal vector to the non-mortar contact interface γ_c^1 at node A. Thus, the contact potential energy Π_{cont} of Eq. (4.10) is re-written as follows,

$$\Pi_{\text{cont}} = \sum_{A=1}^{n^1} \lambda_{NA} g_{NA}. \tag{4.14}$$

The general solution of the unilateral frictionless contact problem obtained by a mortar formulation is given by

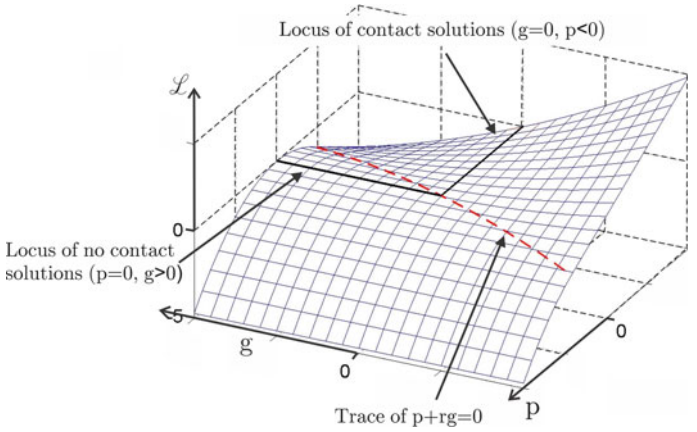


Fig. 4.2 Augmented Lagrangian function for the contact problem. The locus of solutions is displayed

$$
\begin{aligned}
&(\boldsymbol{U}, \boldsymbol{\lambda}) = \arg\min \left(\Pi_{\text{int,ext}} + \Pi_{\text{cont}}\right) \\
&\text{such that for } A = 1, \ldots n^1 \\
&\begin{cases}
\mathrm{g}_{NA} \geq 0 & \text{Gap equation} \\
\lambda_{NA} \leq 0 & \text{Contact force equation} \\
\mathrm{g}_{NA}\lambda_{NA} = 0 & \text{Complementarity condition}
\end{cases}
\end{aligned}
\tag{4.15}
$$

The set of inequalities in Eq. (4.15) establishes the Karush-Kuhn-Tucker conditions (KKT) for the unilateral frictionless contact problem. These inequality constraints can be equivalently written as a sub-differential inclusion as follows:

$$
\lambda_{NA} \in \partial \Psi_{\boldsymbol{R}_+}(\mathrm{g}_{NA}), \tag{4.16}
$$

where $\Psi_{\boldsymbol{R}_+}$ is the indicator function of the real half line $\boldsymbol{R}_+$ and $\partial\Psi_{\boldsymbol{R}_+}$ is the sub-differential of $\Psi_{\boldsymbol{R}_+}$. Equation (4.16) expresses the unilateral contact conditions, with a contact pressure field derived from a non-smooth potential $\Psi_{\boldsymbol{R}_+}(\mathrm{g}_{NA})$ [1, 21]. An augmented Lagrangian function which replaces the energy of contact is defined by

$$
\mathscr{L}_{\text{cont}}(\boldsymbol{U}, \boldsymbol{\lambda}) = \sum_{A=1}^{m^1} \left(\mathrm{g}_{NA} k \lambda_{NA} + \frac{r}{2} \|\mathrm{g}_{NA}(\boldsymbol{U})\|^2 - \frac{1}{2r} \text{dist}^2 \left[k\lambda_{NA} + r\mathrm{g}_{NA}(\boldsymbol{U}), R^- \right] \right), \tag{4.17}
$$

where $\text{dist}(x, C)$ is the distance between x and C, r is a positive penalty parameter and k is a positive scale factor. This function is a C^1 differentiable saddle-point function, as shown in Fig. 4.2. The solution is obtained as the set of values that renders this function stationary. The solution does not depend on the value of parameters r, k. Nevertheless, the convergence rate does depend on these values. The penalty and scale factors, r and k, are selected in terms of a mean value of the Young modulus

E_{mean} and a mean value of the elements size h_{mean} as $10E_{\text{mean}}/h_{\text{mean}}$ to get optimal convergence properties [8].

4.3.1 Mortar Contact Element Definition

A *contact element* is defined for each pair of facets, one on the non-mortar and the other on the mortar surface. If $N1$ is the number of facets on the surface γ_c^1, and $N2$ is the number of facets on the surface γ_c^2, a total of $N1 \times N2$ contact elements are built. Note, however, that only a few of them are active at a given time (i.e. only those elements whose facets are mutually seeing each other). At each contact element, the restrictions to the element facets of the integrals needed for the computation of the weight factors n_{AB}^1 and n_{AB}^2 are evaluated. The generalized coordinates of the contact element are

$$\boldsymbol{\Phi}^e = \left[\boldsymbol{x}_1^{1\,T}\ \boldsymbol{x}_2^{1\,T}\ \ldots\ \boldsymbol{x}_{m^1}^{1\ T}\ \boldsymbol{x}_1^{2\,T}\ \boldsymbol{x}_2^{2\,T}\ \ldots\ \boldsymbol{x}_{m^2}^{2\ T}\ \lambda_{N1}\ \lambda_{N2}\ \ldots\ \lambda_{Nm^1}\right]^T, \quad (4.18)$$

where m^1 and m^2 are the number of nodes of the non-mortar facet and the mortar facet, respectively, $\boldsymbol{x}_I^1$ are the nodal coordinates of the non-mortar facet; $\boldsymbol{x}_I^2$ are the nodal coordinates of the mortar facet, and λ_{NI} are the contact nodal pressures. The number of degrees of freedom of the contact element is $4m^1 + 3m^2$.

4.3.2 Internal Force Vector and Hessian Matrix for Frictionless Contact Problems

The internal forces vector of the contact element is obtained taking variations of $\mathscr{L}_{\text{cont}}^e$, (see Eq. 4.17), with respect to the generalized coordinates $\delta\boldsymbol{\Phi}^e$ as follows

$$\delta\mathscr{L}_{\text{cont}}^e = \delta\boldsymbol{\Phi}^e \cdot \boldsymbol{F}_{\text{cont}}^e(\boldsymbol{\Phi}^e) = \begin{bmatrix} \delta\boldsymbol{x}_B^1 \\ \delta\boldsymbol{x}_C^2 \\ \delta\lambda_{NA} \end{bmatrix} \cdot \sum_{A=1}^{m^1} \begin{bmatrix} n_{AB}^1 \left[\text{proj}_{R^-}(\sigma_{NA})\right] \boldsymbol{\nu}_A \\ -n_{AC}^2 \left[\text{proj}_{R^-}(\sigma_{NA})\ \boldsymbol{\nu}_A\right] \\ -\frac{1}{r}\left[k\lambda_{NA} - \text{proj}_{R^-}(\sigma_{NA})\right] \end{bmatrix}, \quad (4.19)$$

where $\sigma_{NA} = k\lambda_{NA} + r\,g_{NA}$ is a linear combination of primal and dual variables, and $\text{proj}_{R^-}(x)$ is the operator projection of x on R^-. From the definition of the operator $\text{proj}_{R^-}(x)$, it is possible to check that the contribution from node A to the internal force vector of the contact element $\boldsymbol{F}_{\text{cont}}^{e\,A}$ is given by,

$$\delta\boldsymbol{\Phi}^e \cdot \boldsymbol{F}^{e\,A}_{\text{cont}}(\boldsymbol{\Phi}^e) = \begin{bmatrix} \delta\boldsymbol{x}^1_B \\ \delta\boldsymbol{x}^2_C \\ \delta\lambda_{NA} \end{bmatrix} \cdot \begin{cases} \begin{bmatrix} \sigma_{NA}\ \boldsymbol{v}_A\ n^1_{AB} \\ -\sigma_{NA}\ \boldsymbol{v}_A\ n^2_{AC} \\ k\mathrm{g}_{NA} \end{bmatrix}, & \text{if } \sigma_{NA} \le 0, \\ \begin{bmatrix} 0 \\ 0 \\ -\dfrac{k^2}{r}\lambda_{NA} \end{bmatrix}, & \text{if } \sigma_{NA} > 0. \end{cases} \tag{4.20}$$

Contact status is established depending on the sign of σ_{NA}. If $\sigma_{NA} \le 0$, contact occurs; otherwise, if $\sigma_{NA} > 0$, the bodies are separated.

Equilibrium is obtained by solving the following system of nonlinear equations

$$\boldsymbol{G}(\boldsymbol{U}) + \boldsymbol{F}_{\text{cont}}(\boldsymbol{\Phi}) = 0, \tag{4.21}$$

where $\boldsymbol{G}(\boldsymbol{U})$ is the nonlinear vector of the internal and external structural forces, and $\boldsymbol{F}_{\text{cont}}(\boldsymbol{\Phi})$ is the set of contact forces at the interface γ_c, which is obtained by assembling all the contact element contributions $\boldsymbol{F}^e_{\text{cont}}$. The system of Eq. (4.21) does not change during the iterations. This system is solved simultaneously for the displacement and Lagrange multipliers using a standard Newton-Raphson iterative monolithic scheme, and it is not necessary to use any special algorithm for activation/deactivation of the constraints.

The linearization of Eq. (4.20) leads to the contribution of node A to the tangent Hessian matrix of the contact element, which depends on the contact status :

$$\delta\boldsymbol{\Phi}^e \cdot \Delta\boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} \delta\boldsymbol{x}^1_B \\ \delta\boldsymbol{x}^2_C \\ \delta\lambda_B \end{bmatrix} \cdot \begin{cases} \begin{bmatrix} (k\Delta\lambda_A + r\Delta g_{NA})\boldsymbol{v}_A\ n^1_{AB} \\ \quad + \sigma_{NA}(\boldsymbol{v}_A\ \Delta n^1_{AB} + n^1_{AB}\Delta\boldsymbol{v}_A) \\ -(k\Delta\lambda_A + r\Delta \mathrm{g}_{NA})\boldsymbol{v}_A\ n^2_{AC} \\ \quad - \sigma_{NA}(\boldsymbol{v}_A\ \Delta n^2_{AC} + n^2_{AC}\Delta\boldsymbol{v}_A) \\ k\Delta g_{NA}\boldsymbol{\delta}_{AB} \end{bmatrix}, & \sigma_{NA} \le 0 \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\dfrac{k^2}{r}\Delta\lambda_A\boldsymbol{\delta}_{AB} \end{bmatrix}, & \sigma_{NA} > 0. \end{cases} \tag{4.22}$$

The explicit expression of the Hessian matrix is obtained by evaluating the first derivatives of the weight factors n^α_{AB}, the normal interpenetration g_{NA}, and the normal vector $\boldsymbol{v}_A$. A detailed explanation of the linearization of n^α_{AB}, g_{NA} and $\boldsymbol{v}_A$ with the explicit expression of Hessian matrix is given elsewhere in [8].

4.4 Augmented Lagrangian Formulation for Frictional Contact Problems

When friction is considered, the gap vector g_A and the Lagrange multiplier $\boldsymbol{\lambda}_A$ are split into the normal components: g_{NA} and λ_{NA}, and the tangential components: g_{TA} and $\boldsymbol{\lambda}_{TA}$, respectively. The splitting is carried out using the normal vector $\boldsymbol{v}_A$ to the surface γ_c^1 at the node A. Then, the contact potential energy is divided into a normal component and a tangential component yielding

$$\begin{aligned}\Pi_{\text{cont}} = &\sum_{A=1}^{n^1} \underbrace{\left[\lambda_{NA}(t_{n+1})\boldsymbol{v}_A(t_n)\cdot\left(\sum_{B=1}^{n^1} n_{AB}^1(t_{n+1})\,\boldsymbol{x}_B^1(t_{n+1}) - \sum_{C=1}^{n^2} n_{AC}^2(t_{n+1})\,\boldsymbol{x}_C^2(t_{n+1})\right)\right]}_{\text{Normal component}} \\ &+ \sum_{A=1}^{n^1} \underbrace{\boldsymbol{\lambda}_{TA}(t_{n+1})\left[\boldsymbol{I} - \boldsymbol{v}_A(t_n)\otimes\boldsymbol{v}_A(t_n)\right]\cdot\left(\sum_{B=1}^{n^1} n_{AB}^1(t_{n+1})\,\boldsymbol{x}_B^1(t_n) - \sum_{C=1}^{n^2} n_{AC}^2(t_{n+1})\,\boldsymbol{x}_C^2(t_n)\right)}_{\text{Tangencial component}}.\end{aligned} \tag{4.23}$$

Note that in the term corresponding to the normal component, the weight factors n_{AB}^{α} and the positions $\boldsymbol{x}_A^{\alpha}$ are evaluated at the current time step t_{n+1}, while in the term corresponding to the tangential component, the positions are evaluated at the previous time step t_n. In what follows, the time variable t will be omitted in the equations in order to simplify the notation. This definition of the tangential component allows the representation of the incremental tangential movement ensuring objectivity properties in the formulation, as demonstrated by Puso and Laursen [39]. The normal vector $\boldsymbol{v}_A$ is the average normal to the facets touching node A. In order to simplify the formulation, $\boldsymbol{v}_A$ is evaluated at the previous time step t_n; thus, it does not contribute to the Hessian matrix.

The general solution to the unilateral frictional contact problem by a mortar formulation is then given by,

$$\begin{aligned}&(\boldsymbol{U},\boldsymbol{\lambda}) = \arg\min\left(\Pi_{\text{int,ext}} + \Pi_{\text{cont}}\right) \\ &\text{such that for } A = 1,\ldots n^1 \\ &\begin{cases} \mathrm{g}_{NA} \geq 0 & \text{Gap equation} \\ \lambda_{NA} \leq 0 & \text{Contact force equation} \\ \mathrm{g}_{NA}\lambda_{NA} = 0 & \text{Complementarity condition} \end{cases} \\ &\begin{cases} \|\boldsymbol{\lambda}_{TA}\| \leq -\mu\lambda_{NA} & \text{Friction law} \\ \dot{g}_{TA} = \|\dot{g}_{TA}\|\,\boldsymbol{\lambda}_{TA}/\|\boldsymbol{\lambda}_{TA}\| & \text{Slip rule} \\ \|g_{TA}\|\left(\|\boldsymbol{\lambda}_{TA}\| + \mu\lambda_{NA}\right) = 0 & \text{Complementarity condition} \end{cases}\end{aligned} \tag{4.24}$$

The second set of equations represents the KKT conditions for the frictional contact problem, where μ is the friction coefficient and $\dot{g}_{TA}$ is the relative velocity vector between bodies. The last set of constraints in Eq. (4.24) referring to the contact friction problem, is equivalent to the following sub-differential inclusion,

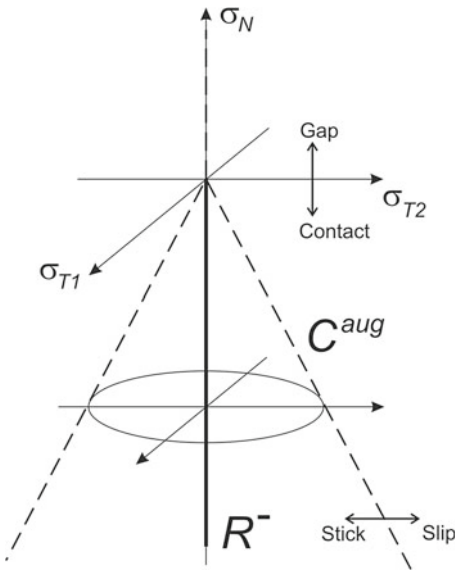


Fig. 4.3 Augmented friction *cone* $C(\lambda_N + r g_N)$

$$\boldsymbol{\lambda}_{TA} \in \partial \Psi_C(g_{TA}), \tag{4.25}$$

where $\partial \Psi_C$ is the indicator function of the convex disc C with radius $-\mu\lambda_{NA}$ centered at the origin:

$$C(\lambda_{NA}) = \frac{\boldsymbol{\lambda}_{TA}}{\|\boldsymbol{\lambda}_{TA}\|} \leq -\mu\lambda_{NA}, \tag{4.26}$$

This set of equations represents the isotropic friction Coulomb law. The problem stated in Eq. (4.24) can be regularized using a dual-mixed formulation based on an augmented Lagrangian, as proposed in the work of Alart and Curnier [1],

$$\mathscr{L}_{\text{cont}}(\boldsymbol{U}, \boldsymbol{\lambda}) = \sum_{A=1}^{m^1} \Big(\mathrm{g}_{NA}\lambda_{NA} + \frac{r}{2}\|\mathrm{g}_{NA}(\boldsymbol{U})\|^2 - \frac{1}{2r}\mathrm{dist}^2\left[\lambda_{NA} + r\,\mathrm{g}_{NA}(\boldsymbol{U}), R^-\right]$$
$$g_{TA} \cdot \boldsymbol{\lambda}_{TA} + \frac{r}{2}\|g_{TA}(\boldsymbol{U})\|^2 - \frac{1}{2r}\mathrm{dist}^2\left[\boldsymbol{\lambda}_{TA} + r g_{TA}(\boldsymbol{U}), C^{\text{aug}}\right]\Big). \tag{4.27}$$

Here, C^{aug} is the convex set of friction $C(\lambda_{NA} + r\mathrm{g}_{NA})$. The use of $C(\lambda_{NA} + r\mathrm{g}_{NA})$ corresponds to extending the friction cone defined over the augmented Lagrange multiplier, to the positive half line; consequently, a Lagrangian function whose stationary values give the solution to the frictional contact problem, is obtained (Fig. 4.3).

4.4.1 Mortar Contact Element Definition for Frictional Contact Problems

The contact force $\boldsymbol{\lambda}_A$ acting in the body Ω^1 is referred to a local frame $\boldsymbol{A} = (\boldsymbol{v}_A, \boldsymbol{t}_{A_1}, \boldsymbol{t}_{A_2})$ at a node A, as depicted in Fig. 4.1. Then, the contact force $\boldsymbol{\lambda}_A$ is split into the normal and the tangential components by using the normal vector $\boldsymbol{v}_A$ of the frame $\boldsymbol{A}$, yielding

$$\begin{aligned} \lambda_{NA} &= \boldsymbol{\lambda}_A \cdot \boldsymbol{v}_A, \\ \boldsymbol{\lambda}_{TA} &= \boldsymbol{\lambda}_A - \lambda_{NA} \boldsymbol{v}_A. \end{aligned} \tag{4.28}$$

Finally, to simplify the notation and the interpretation of the contact status, the augmented Lagrange multiplier is given by,

$$\boldsymbol{\sigma}_A = \sigma_{NA} \boldsymbol{v}_A + \boldsymbol{\sigma}_{TA}, \tag{4.29}$$

where

$$\begin{aligned} \sigma_{NA} &= k\lambda_{NA} + r\mathrm{g}_{NA}, \\ \boldsymbol{\sigma}_{TA} &= k\boldsymbol{\lambda}_{TA} + r g_{TA}. \end{aligned} \tag{4.30}$$

The generalized coordinates of the contact element are

$$\boldsymbol{\Phi}^e = \left[\boldsymbol{x}_1^{1\,T} \; \boldsymbol{x}_2^{1\,T} \; \ldots \; \boldsymbol{x}_{m^1}^{1\,T} \; \boldsymbol{x}_1^{2\,T} \; \boldsymbol{x}_2^{2\,T} \; \ldots \; \boldsymbol{x}_{m^2}^{2\,T} \; \boldsymbol{\lambda}_1 \; \boldsymbol{\lambda}_2 \; \ldots \; \boldsymbol{\lambda}_{m^1} \right]^T, \tag{4.31}$$

where m^1 and m^2 are the number of nodes of the non-mortar facet and the mortar facet, respectively; $\boldsymbol{x}_I^1$ are the nodal coordinates of the non-mortar facet; $\boldsymbol{x}_I^2$ are the nodal coordinates of the mortar facet; and $\boldsymbol{\lambda}_I$ are the contact nodal pressures. The number of degrees of freedom of the contact element is $6m^1 + 3m^2$. The size of the system of equations is increased with respect to the frictionless case due to the consideration of tangential contact pressures.

4.4.2 Internal Force Vector and Hessian Matrix for Frictional Contact Problems

The internal force vector for a contact element is obtained by taking variations in $\mathscr{L}_{\text{cont}}$, see Eq. (4.27); thus,

$$\delta \mathscr{L}_{\text{cont}} = \begin{bmatrix} \delta \boldsymbol{x}_B^1 \\ \delta \boldsymbol{x}_C^2 \\ \delta \boldsymbol{\lambda}_A \end{bmatrix} \cdot \sum_{A=1}^{m^1} \begin{bmatrix} n_{AB}^1 \left[\mathrm{proj}_{R^-}(\sigma_{NA}) \; \boldsymbol{v}_A + \mathrm{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) \right] \\ -n_{AC}^2 \left[\mathrm{proj}_{R^-}(\sigma_{NA}) \; \boldsymbol{v}_A + \mathrm{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) \right] \\ -\frac{k}{r} \left[k\lambda_{NA} - \mathrm{proj}_{R^-}(\sigma_{NA}) \right] \boldsymbol{v}_A - \frac{k}{r} \left[k\boldsymbol{\lambda}_{TA} - \mathrm{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) \right] \end{bmatrix}. \tag{4.32}$$

The projection operator $\text{proj}_C(x)$ is defined as

$$\text{proj}_C(x) = \begin{cases} x, & \text{if} \quad \|x\| \leq \varepsilon, \\ \epsilon x/\|x\|, & \text{if} \quad \|x\| > \varepsilon, \end{cases} \tag{4.33}$$

where the scalar value ϵ corresponds to the radius of the disk in the friction Coulomb cone. The contact status of *gap*, *stick* or *slip* can then be determined:

- The condition $\sigma_{NA} \geq 0$ is associated to the *gap* condition, in this case, $\text{proj}_{R^-}(\sigma_{NA}) = 0$ and $\text{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) = 0$. For this reason, from Eq. (4.32), the contribution of node A to the internal contact force vector yields,

$$\boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} 0 \\ 0 \\ -\frac{k^2}{r}\boldsymbol{\lambda}_A \end{bmatrix}. \tag{4.34}$$

- The *stick* condition is obtained when $\|\boldsymbol{\sigma}_{TA}\| < -\mu\sigma_{NA}$, in this case, $\text{proj}_{R^-}(\sigma_{NA}) = \sigma_{NA}$ and $\text{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) = \boldsymbol{\sigma}_{TA}$. Therefore, from Eq. (4.32), the contribution of node A to the internal contact force vector yields,

$$\boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} n^1_{AB}\boldsymbol{\sigma}_A \\ -n^2_{AC}\boldsymbol{\sigma}_A \\ k g_A \end{bmatrix}. \tag{4.35}$$

- Finally, the *slip* condition is $\|\boldsymbol{\sigma}_{TA}\| \geq -\mu\sigma_{NA}$, in this case, $\text{proj}_{R^-}(\sigma_{NA}) = \sigma_{NA}$ and $\text{proj}_{C^{\text{aug}}}(\boldsymbol{\sigma}_{TA}) = -\mu\sigma_{NA}\boldsymbol{t}_A$. Thus, from Eq. (4.32), the contribution of node A to the internal contact force vector yields,

$$\boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} n^1_{AB}(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) \\ -n^2_{AC}(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) \\ k\boldsymbol{v}_A g_{NA} - \frac{k}{r}(k\boldsymbol{\lambda}_{TA} + \mu\sigma_{NA}\boldsymbol{t}_A) \end{bmatrix}. \tag{4.36}$$

  Here, $\boldsymbol{t}_A$ is calculated from the following definition,

$$\boldsymbol{t}_A = \frac{\boldsymbol{\sigma}_{TA}}{\|\boldsymbol{\sigma}_{TA}\|}, \tag{4.37}$$

  which represents the tangential unit vector in the slip direction at node A.

The linearization of Eq. (4.20) leads to the tangent Hessian matrix of the contact element, which depends on the contact status

- *Gap* status $\sigma_{NA} \geq 0$

$$\Delta\boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} 0 \\ 0 \\ -\frac{k^2}{r}\Delta\boldsymbol{\lambda}_A \end{bmatrix}. \tag{4.38}$$

- *Stick* status $\|\boldsymbol{\sigma}_{TA}\| < -\mu\sigma_{NA}$

$$\Delta \boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} \Delta n^1_{AB}\boldsymbol{\sigma}_A + n^1_{AB}\Delta\boldsymbol{\sigma}_A \\ -\Delta n^2_{AC}\boldsymbol{\sigma}_A - n^2_{AC}\Delta\boldsymbol{\sigma}_A \\ k\Delta g_A \end{bmatrix}. \tag{4.39}$$

- *Slip* status $\|\boldsymbol{\sigma}_{TA}\| \geq -\mu\sigma_{NA}$

$$\Delta \boldsymbol{F}^{e\,A}_{\text{cont}} = \begin{bmatrix} \Delta n^1_{AB}\,(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) + n^1_{AB}\,\Delta(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) \\ -\Delta n^2_{AC}\,(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) - n^2_{AC}\,\Delta(\boldsymbol{v}_A - \mu\boldsymbol{t}_A\sigma_{NA}) \\ \Delta\left(k\boldsymbol{v}_A g_{NA} - \frac{k}{r}(k\boldsymbol{\lambda}_{TA} + \mu\sigma_{NA}\boldsymbol{t}_A)\right) \end{bmatrix}. \tag{4.40}$$

The linearization of these quantities can be calculated from the work of Cavalieri and Cardona [8], except for the linearization of the tangential vector $\boldsymbol{t}_A$ which is outlined in the `Appendix`. Here, it is important to remark that since $\boldsymbol{v}_A$ is evaluated at the previous time step, its linearization does not contribute to the Hessian matrix. This assumption leads to a simple formulation, with good convergence properties of the nonlinear problem. As it will be shown in the examples, this assumption does not impose a severe restriction. However, we remark that the linearization of the average normal $\boldsymbol{v}_A$ at the current time step could have been incorporated without important modifications to the formulation and to the code.

4.5 Numerical Examples

Three numerical examples including finite deformation and large slip displacement are presented to evaluate the robustness and accuracy of the proposed contact algorithm. The examples involve quasi-static simulations and were carried-out in the research finite element code Oofelie [6], where the contact algorithm is integrated. All pre- and post-processing tasks where performed using the SAMCEF-Field software.

4.5.1 Validation Example I: The Contact Patch Test

The first example is a contact patch test proposed by Chen and Hisada [10]. The 3D solutions obtained in this work are compared with the 2D solutions of Chen and Hisada, introducing a plane strain state which reproduces the same boundary conditions. The material behavior used in this example is linear elastic. The mesh topology, boundary conditions, dimensions and material properties are shown in Fig. 4.4a. Figure 4.4b shows that the stress field is transmitted exactly from one body to the other with non-conforming meshes, concluding that the formulation verifies the contact patch test to machine accuracy. When using a node-segment

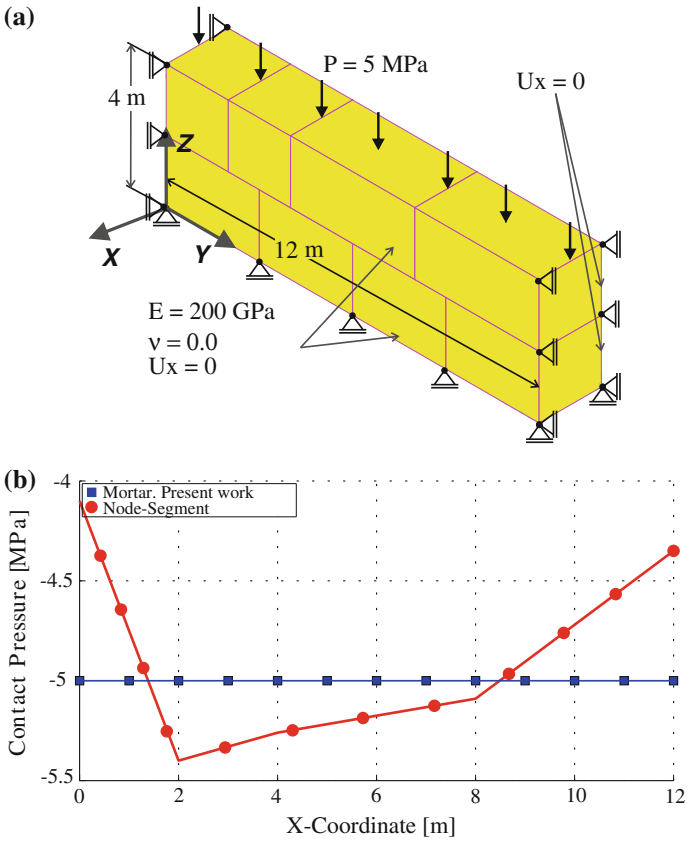


Fig. 4.4 Contact patch test. **a** Boundary conditions and mechanical properties. **b** Stress solution of the node-segment and mortar algorithms in the contact interface

approach, the stress field is not uniform, and thus, it does not pass the contact patch test (see Fig. 4.4b). In order to study the influence of the parameters r and k on the convergence properties, solutions for different values of r and k were computed. From the geometry and material properties depicted in Fig. 4.4, the recommended value results:

$$r = k = 10\frac{E}{h} = 10\frac{2.1 \times 10^9}{2} = 1.05 \times 10^{10} \tag{4.41}$$

Table 4.1 gives the residual norm per iteration for the different values of r and k. Quadratic convergence behavior is displayed in almost all cases. The condition number of the global tangent matrix is shown in Table 4.1. We see that the minimum condition number is obtained for values of r and k close to the value suggested by $10E_{\text{mean}}/h_{\text{mean}}$.

Table 4.1 Residual norm evolution in the contact patch test problem for different values of k and r ($r = k$)

Iteration	10^6	10^8	10^{10}	10^{12}	10^{14}
1	1.42×10^{-2}	3.35×10^{-1}	10.2751	84.1781	5.43×10^{3}
2	3.07×10^{-1}	4.84×10^{-2}	7.80×10^{-2}	6.99×10^{-1}	9.14×10^{1}
3	1.75×10^{-3}	7.39×10^{-3}	3.89×10^{-6}	1.07×10^{-3}	3.37×10^{3}
4	1.24×10^{-4}	7.78×10^{-5}		1.39×10^{-9}	2.77×10^{1}
5	3.77×10^{-11}	2.18×10^{-11}			1.93×10^{3}
6					1.24×10^{2}
7					8.32×10^{2}
8					3.84×10^{5}
9					FAIL

4.5.2 Validation Example II

This test represents an important validation example to study frictional contact algorithms. The example has been proposed first by Oden and Pire [31] as a 2D example, whereas more recent solutions can be found in [2, 3, 7] by using a node-to-segment approach, or in the work of Fischer and Wriggers using a 2D mortar approach [15]. In this work, three-dimensional solutions have been computed introducing plane strain boundary conditions to reproduce the same results as in the work of Armero and Petocz [3]. The mesh topology, boundary conditions and mechanical properties used in the simulation are shown in Fig. 4.5. The material behavior is linear elastic. The upper block has a mesh with 462 nodes and 200 hexaedric elements. The contact zone has a longitudinal distance of 3.6, as shown in Fig. 4.5. A uniform pressure $q_z = 200$ is applied to the top surface of the upper body, producing a deformation against the rigid foundation. Then, another pressure field $q_x = 60$ is actuating on one side of the body pushing it in the X direction. Figure 4.6a shows a numerical comparison of the normal and tangential stress in the contact interface. Both solutions shows agreement with Armero and Petocz [3]. Figure 4.6b shows a quadratic residue evolution.

4.5.3 Validation Example III

The test presented in this section, is based on a bidimensional example proposed by Hammer [17]. The boundary conditions for a 3D version of the Hammer problem are introduced to represent a plane strain state. The dimensions and mesh topology are shown in Fig. 4.7. The mesh has 100 nodes and 39 hexahedral linear finite elements as shown in Fig. 4.7. The material behaviour for both bodies is Neo-Hookean with the stored energy density expressed as

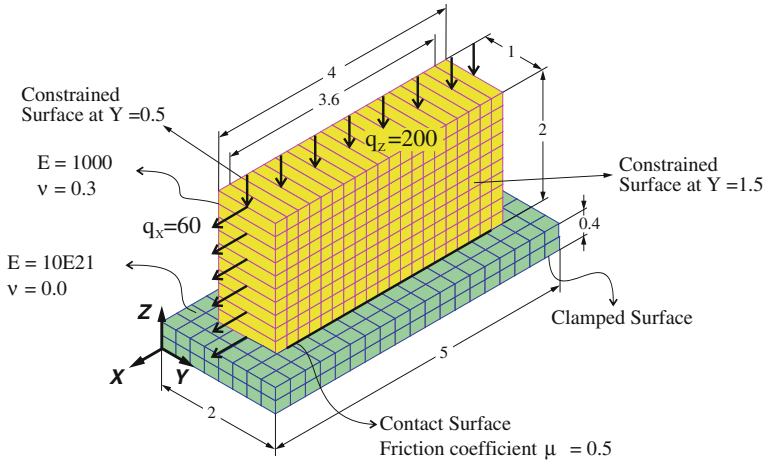


Fig. 4.5 Elastic body pressured with a rigid surface and pulled tangentially

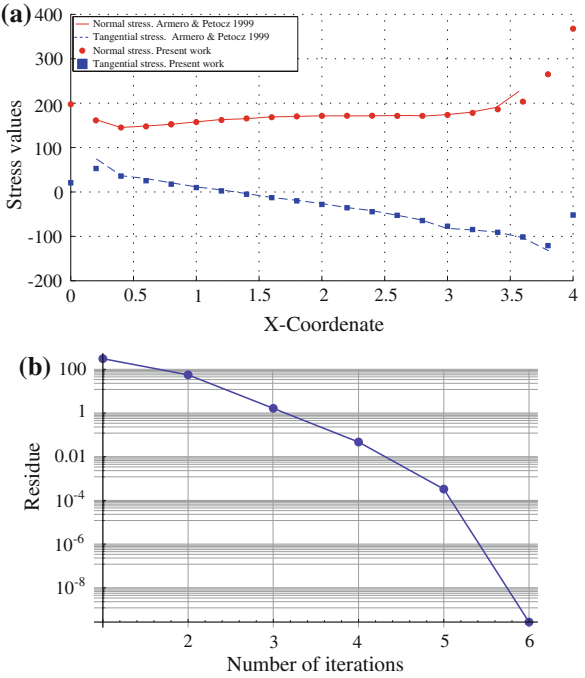


Fig. 4.6 Numerical solutions for the validation example II. **a** Normal and tangential stress variation at the contact interface. Numerical solutions compared with [3]. **b** Residue evolution

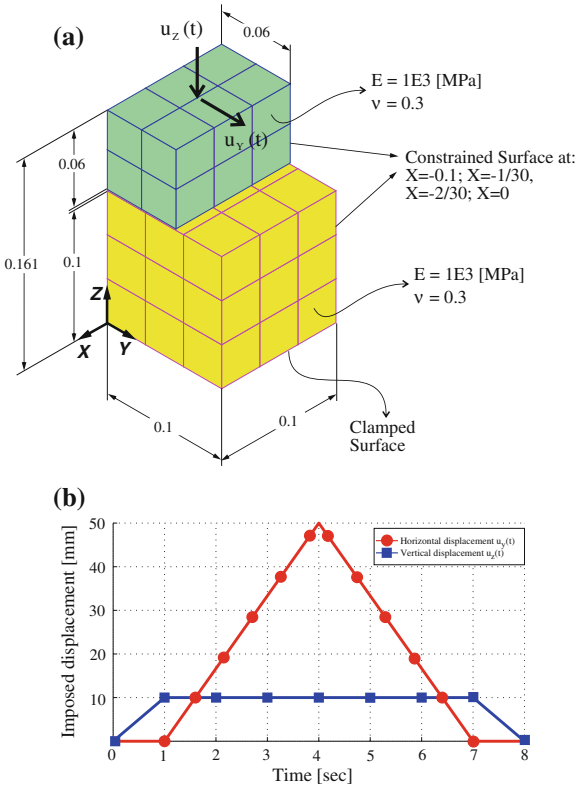


Fig. 4.7 Validation example III. **a** Mechanical properties and boundary conditions. **b** Imposed displacements on the top side of the block

$$W(I, J) = \frac{\mu}{2}(I - 3 - 2\ln J) + \frac{\lambda}{2}(J - 1)^2, \tag{4.42}$$

where λ and μ are the first and second Lamé parameters, respectively. Figure 4.7a shows the mechanical properties used in the example. At the beginning of the simulation, there is an initial gap of 1 mm between the bodies. This gap is introduced to demonstrate that the implementation is able to capture new contact surfaces, see Fig. 4.8a. The upper block is moved downwards in the vertical direction with a movement law as shown in Fig. 4.7b. At time 1 s, the vertical displacement $u_z(t)$ arrives at 10 mm. At this moment, the deformation state is shown in Fig. 4.8b. After time 1.6 s, the upper body starts to slip and change of contact status, from stick to slip. Fig. 4.8c shows the deformation of both blocks due to friction forces. Then, the upper block is moved in the Y direction with an horizontal displacement u_y, until arriving at 50 mm at time step 4 s, see Fig. 4.8d. After time 4 s the movement direction changes and continues until time 7 s. Here, the upper block starts lifting to the initial position reached at time 8 s. The total horizontal and vertical reactions forces evaluated on the

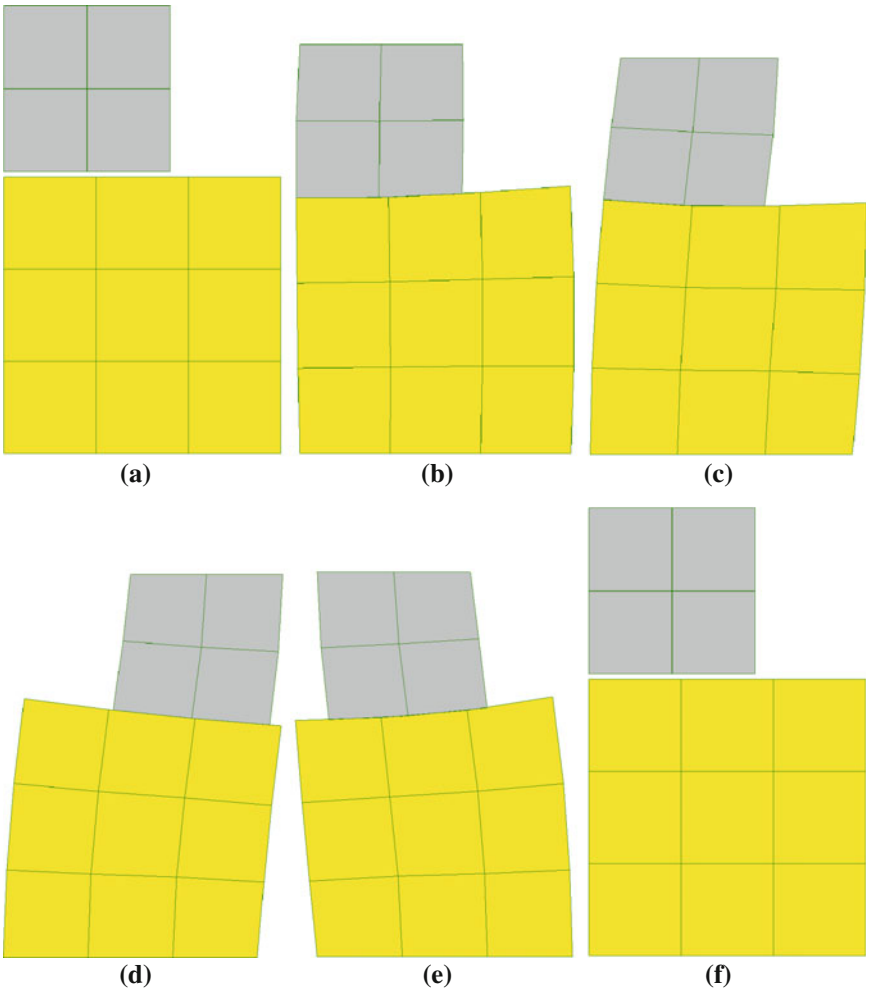


Fig. 4.8 Validation example III. Deformation evolution. **a** Time: 0 s. **b** Time: 1 s. **c** Time: 1.7 s. **d** Time: 4 s. **e** Time: 7 s. **f** Time: 8 s

top side of the upper block is compared with the example proposed by Hammer [17]. Both solutions are presented in Fig. 4.9a showing agreement. Figure 4.9b shows a quadratic residue evolution.

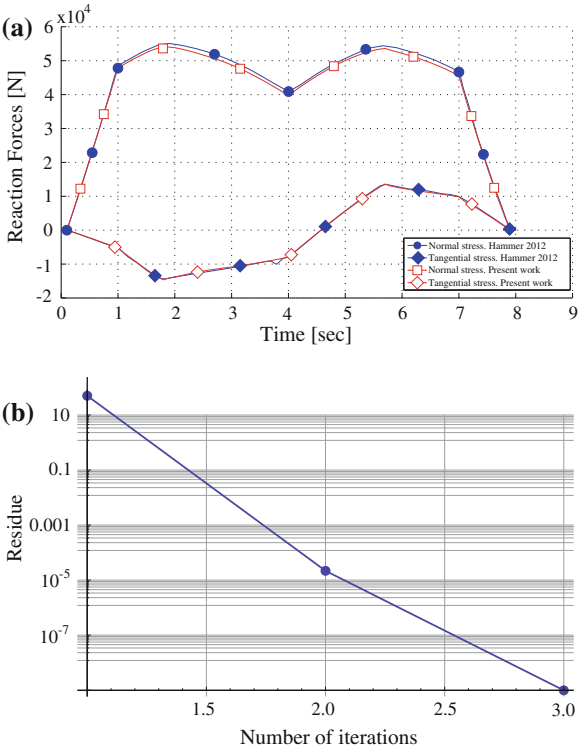


Fig. 4.9 Numerical solutions for the validation example III. **a** Horizontal and vertical reaction forces evaluated on the top side of the upper block. **b** Residue evolution

4.6 Conclusions

In this work, a general contact algorithm using an augmented Lagrangian method is presented. The algorithm has three main features: it allows to solve the equations using a semismooth Newton method avoiding the programming complications of algorithms based on activation/deactivation of constraints with a two-leveled iterative loop; it is applicable to problems with non-conforming meshes; and the user does not need to define any penalty parameter. The equations for the computation of the residual forces and tangent matrices were presented. The numerical examples that have been presented, demonstrated the ability of the scheme to represent frictionless and frictional contact problems with small and large displacements. Finally, in the presented examples, quadratic convergence rate with a small number of iterations is achieved.

Acknowledgments This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT), Universidad Nacional del Litoral (CAI+D 2009 PI65-330).

Appendix

The linearization of the tangential vector $\boldsymbol{t}_A$ is presented. The tangential vector $\boldsymbol{t}_A$ is used in the slip status, thus

$$\boldsymbol{t}_A = \frac{\boldsymbol{\sigma}_{TA}}{\| \boldsymbol{\sigma}_{TA} \|} = \frac{\boldsymbol{\sigma}_{TA}}{-\mu\sigma_{NA}}. \tag{4.43}$$

The linearization operator Δ applied to Eq. (4.43), yields

$$\Delta\boldsymbol{t}_A = \frac{[\boldsymbol{I} - \boldsymbol{t}_A \otimes \boldsymbol{t}_A]\Delta\boldsymbol{\sigma}_A}{\| \boldsymbol{\sigma}_{TA} \|} = \frac{[\boldsymbol{I} - \boldsymbol{t}_A \otimes \boldsymbol{t}_A]\Delta\boldsymbol{\sigma}_A}{-\mu\sigma_{NA}}. \tag{4.44}$$

After some algebraic manipulations the linearization of the tangential vector is written as

$$\Delta\boldsymbol{t}_A = -\frac{\boldsymbol{I} - \boldsymbol{t}_A \otimes \boldsymbol{t}_A - \boldsymbol{v}_A \otimes \boldsymbol{v}_A}{\mu\sigma_{NA}}\Delta\boldsymbol{\sigma}_A + \frac{\boldsymbol{v}_A \otimes \boldsymbol{\sigma}_A + (\boldsymbol{I} - \boldsymbol{t}_A \otimes \boldsymbol{t}_A)\sigma_{NA}}{\mu\sigma_{NA}}\Delta\boldsymbol{v}_A. \tag{4.45}$$

If the variation of the normal vector $\boldsymbol{v}_A$ is neglected, the final expression is given by

$$\Delta\boldsymbol{t}_A = -\frac{\boldsymbol{I} - \boldsymbol{t}_A \otimes \boldsymbol{t}_A - \boldsymbol{v}_A \otimes \boldsymbol{v}_A}{\mu\sigma_{NA}}\Delta\boldsymbol{\sigma}_A. \tag{4.46}$$

References

1. Alart P, Curnier A (1991) A mixed formulation for frictional contact problems prone to newton like solution methods. Comput Methods Appl Mech Eng 92(3):353–375
2. Areias PMA, César de Sá JMA, Conceiçao António CA (2004) Algorithms for the analysis of 3D finite strain contact problems. Int J Numer Methods Eng 61:1107–1151
3. Armero F, Petocz E (1999) A new dissipative time-stepping algorithm for frictional contact problems: formulation and analysis. Comput Methods Appl Mech Eng 179:151–178
4. Bernardi C, Maday Y, Patera A (1992) A new nonconforming approach to domain decomposition: the mortar element method. In: Brezia JLEH (ed) Nonlinear partial differential equations and their applications, Pitman and Wiley, pp 13–51
5. Brezzi F, Fortin M (1991) Mixed and hybrid finite element methods. Springer, New York
6. Cardona A, Klapka I, Geradin M (1994) Design of a new finite element programming environment. Eng Comput 11:365–381
7. Cavalieri F, Cardona A (2012) An augmented Lagrangian method to solve 3D contact problems. Latin Am Appl Res 42(201):281–289
8. Cavalieri F, Cardona A (2013) An augmented Lagrangian technique combined with a mortar algorithm for modelling mechanical contact problems. Int J Numer Methods Eng 93(4):420–442
9. Cavalieri F, Fachinotti V, Cardona A (2012) A mortar contact algorithm for three-dimensional elasticity problems. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería 28(2):80–92
10. Chen X, Hisada T (2006) Development of a finite element contact analysis algorithm to pass the patch test. Jpn Soc Mech Eng 49:483–491

11. Cichosz T, Bischoff M (2011) Consistent treatment of boundaries with mortar contact formulations using dual lagrange multipliers. Comput Methods Appl Mech Eng 200(9–12):1317–1332
12. De Lorenzis L, Temizer I, Wriggers P, Zavarise G (2011) A large deformation frictional contact formulation using NURBS-based isogeometric analysis. Int J Numer Methods Eng 87(13):1278–1300
13. De Lorenzis L, Wriggers P, Zavarise G (2012) A mortar formulation for 3D large deformation contact using nurbs-based isogeometric analysis and the augmented Lagrangian method. Comput Mech 49(1):1–20
14. Fischer K, Wriggers P (2005) Frictionless 2D contact formulations for finite deformations based on the mortar method. Comput Mech 36(3):226–244
15. Fischer K, Wriggers P (2006) Mortar based frictional contact formulation for higher order interpolations using the moving friction cone. Comput Methods Appl Mech Eng 195:5020–5036
16. Gitterle M, Popp A, Gee MW, Wall WA (2010) Finite deformation frictional mortar contact using a semi-smooth newton method with consistent linearization. Int J Numer Methods Eng 84(5):543–571
17. Hammer M (2012) Frictional mortar contact for finite deformation problems with synthetic contact kinematics. Ph.D. thesis, Institut für Festigkeitslehre der Technischen Universitat Graz
18. Hartmann S, Ramm E (2008) A mortar based contact formulation for non-linear dynamics using dual lagrange multipliers. Finite Elem Anal Des 44:245–258
19. Hüeber S, Wohlmuth B (2005) A primal dual active set strategy for non-linear multibody contact problems. Comput Meth Appl Mech Eng 194(27–29):3147–3166
20. Hüeber S, Wohlmuth B (2009) Thermo-mechanical contact problems on non-matching meshes. Comput Methods Appl Mech Eng 198(15–16):1338–1350
21. Heegaard J, Curnier A (1993) An augmented Lagrangian method for discrete large slip contact problems. Int J Numer Methods Eng 36:569–593
22. Heintz P, Hansbo P (2006) Stabilized lagrange multiplier methods for bilateral elastic contact with friction. Comput Methods Appl Mech Eng 195(33–36):4323–4333
23. Hesch C, Betsch P (2010) Transient three-dimensional domain decomposition problems: frame-indifferent mortar constraints and conserving integration. Int J Numer Methods Eng 82(3):329–358
24. Hesch C, Betsch P (2011) Transient three-dimensional contact problems: mortar method. Mixed methods and conserving integration. Comput Mech 48(4):461–475
25. Hüeber S, Mair M, Wohlmuth BI (2005) A priori error estimates and an inexact primal-dual active set strategy for linear and quadratic finite elements applied to multibody contact problems. Appl Numer Math 54(3–4):555–576
26. Jones R, Papadopoulos P (2001) A novel three-dimensional contact finite element based on smooth pressure interpolations. Int J Numer Methods Eng 51(7):791–811
27. Kikuchi N, Oden J (1988) Contact problems in elasticity: a study of variational inequalities and finite element methods. SIAM 1988
28. Kim J, Youn S (2012) Isogeometric contact analysis using mortar method. Int J Numer Methods Eng 89(12):1559–1581
29. Laursen T (2002) Computational contact and impact mechanics. Springer, Berlin
30. McDevitt TW, Laursen TA (2000) A mortar-finite element formulation for frictional contact problems. Int J Numer Methods Eng 48(10):1525–1547
31. Oden J, Pire E (1984) Algorithms and numerical results for finite element approximation of contact problems with non-classical friction laws. Comput Struct 19:137–147
32. Papadopoulos P, Solberg J (1998) A Lagrange multiplier method for the finite element solution of frictionless contact problems. Math Comput Model 28(4–8):373–384
33. Papadopoulos P, Taylor R (1990) A mixed formulation for the finite element solution of contact problems. Technical Report UCB/ SEMM Report 90/18, University of California, Berkeley
34. Popp A, Gee M, Wall WA (2009) A finite deformation mortar contact formulation using a primal dual active set strategy. Int J Numer Methods Eng 79(11):1354–1391

35. Popp A, Gitterle M, Gee MW, Wall W (2010) A dual mortar approach for 3D finite deformation contact with consistent linearization. Int J Numer Methods Eng 83(11):1428–1465
36. Popp A, Seitz A, Gee MW, Wall WA (2013) Improved robustness and consistency of 3D contact algorithms based on a dual mortar approach. Comput Methods Appl Mech Eng 264:67–80
37. Puso M (2004) A 3D mortar method for solid mechanics. Int J Numer Methods Eng 59:315–336
38. Puso M, Laursen T (2004) A mortar segment-to-segment contact method for large deformation solid mechanics. Comput Methods Appl Mech Eng 193:601–629
39. Puso M, Laursen T (2004) A mortar segment-to-segment frictional contact method for large deformations. Comput Methods Appl Mech Eng 193:4891–4913
40. Solberg J, Jones R, Papadopoulos P (2007) A family of simple two-pass dual formulations for the finite element solution of contact problems. Comput Methods Appl Mech Eng 196 (4–6):782–802
41. Solberg J, Papadopoulos P (2005) An analysis of dual formulations for the finite element solution of two-body contact problems. Comput Methods Appl Mech Eng 194(25–26):2734–2780
42. Temizer A (2012) A mixed formulation of mortar-based frictionless contact. Comput Methods Appl Mech Eng 223–224:173–185
43. Temizer A (2013) A mixed formulation of mortar-based contact with friction. Comput Methods Appl Mech Eng 255:183–195
44. Temizer A, Wriggers P, Hughes J (2012) Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. Comput Methods Appl Mech Eng 209–212:115–128
45. Wohlmuth B (2000) A mortar finite element method using dual spaces for the Lagrange multiplier. SIAM J Numer Anal 38:989–1012
46. Wriggers P (2002) Computational contact mechanics. Wiley, New York
47. Yang B, Laursen T, Meng X (2005) Two dimensional mortar contact methods for large deformation frictional sliding. Int J Numer Methods Eng 62:1183–1225

Chapter 5
Contact Dynamics Formulation Using Minimal Coordinates

Abhinandan Jain

Abstract In recent years, complementarity techniques have been developed for solving non-smooth multibody dynamics involving contact and collision events. The linear complementarity approach sets up a linear complementarity problem (LCP) using non-minimal coordinates for the unilateral contact constraints and inter-link bilateral constraints on the system. In this chapter, we develop a complementarity formulation that uses minimal coordinates. This results in a much smaller LCP whose size is independent of the number of bodies and the number of degrees of freedom in the system. Furthermore, we exploit operational space low-order algorithms to overcome key computational bottlenecks to obtain over an order of magnitude speed up in the solution procedure.

5.1 Introduction

For more than a decade, researchers have been developing complementarity based approaches for formulating and solving the equations of motion of multibody systems with contact and collision dynamics [1–3]. Examples of such dynamics for robotic systems include manipulation and grasping tasks such as illustrated in Fig. 5.1, and legged locomotion. The complementarity approach models bodies as rigid, and uses impulsive dynamics to handle non-smooth collision and contact interactions. Complementarity methods impulsively "step" over non-smooth events and thus avoid small integration step sizes encountered with penalty based methods that model surface compliance during contact [4]. In this chapter, we focus on a minimal coordinate formulation of the complementarity approach for contact and collision dynamics for multi-link systems. This chapter builds upon the operational space formulation for contact and collision dynamics described in reference [5] and adopts the linear complementarity based physics models from [2, 3].

A. Jain (✉)
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
e-mail: abhi.jain@jpl.nasa.gov

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_5,

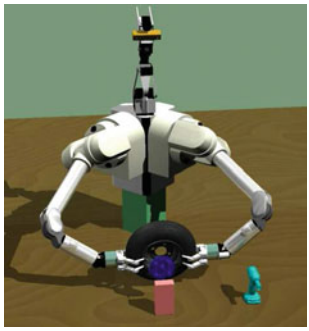


Fig. 5.1 An example multi-arm robot manipulation task involving unmounting a wheel from a hub involving several contact and collision dynamics interaction events

Generally, the complementarity based solution consists of a combination of: (a) setting up a *linear complementarity problem* (*LCP*) problem; (b) numerically solving the LCP; and (c) ancillary dynamics computations. The LCP takes into account the link mass and inertia properties, contact friction parameters, inter-link bilateral constraints and contact and collision unilateral constraints. The LCP solution identifies the unilateral constraints that are active, and solves for the impulsive forces and velocity changes that are consistent with the constraints on the system. Variants of the complementarity approach to handle elastic and inelastic collisions have also been developed [3]. While LCP formulations use discretized approximations for the friction cones, other researchers have explored non-linear cone complementarity approaches that avoid such approximations [6, 7].

The typical approach to handling contact and collision dynamics is to work with non-minimal coordinates, since the LCP is simpler to set up [3]. For a multi-link system with n links, the LCP involves $6n$ non-minimal coordinates, together with the bilateral constraints associated with the inter-link hinges in this approach. The mass matrix is block diagonal and constant. However, the LCP dimension is large and computationally expensive to solve. In addition, these formulations require additional measures for managing error drift in the bilateral constraints when propagating the system dynamics state.

An alternative approach is to use minimal hinge coordinates [8]. While the underlying physics remains unchanged, due to the much smaller number of generalized coordinates, the size of the dynamics model is much smaller. As a consequence the size of the LCP problem is reduced. Also, the bilateral constraints for the inter-link hinges are eliminated along with the need to manage their constraint violation errors. However, the use of minimal coordinates does lead to dense and configuration dependent mass matrices. Thus while minimal coordinates lead to smaller LCP problems, they also significantly increase the difficulty and computational cost of setting up the LCP. This has been a significant hurdle in the use of minimal coordinate approaches.

In this chapter[1] we explore a progression of minimal coordinate formulations that partition the overall solution effort in different ways between setting up the LCP, and

[1] This research on minimal coordinate contact dynamics has also been reported in a recent conference paper [9].

solving it. Our goal is to reduce the overall computational cost by (a) taking advantage of the smaller dimension of minimal coordinate models, and (b) exploiting the host of structure based, and low-order dynamics algorithms that are available for minimal coordinate dynamics models. Notable examples of such structure based algorithms include the composite rigid body algorithms for computing the mass matrix [10], the articulated body inertia forward dynamics algorithm [11] and the spatial operator based operational space dynamics algorithm [12].

The main contribution of this chapter is in the development of an operational space based *OS formulation*, that uses minimal coordinates for the contact and collision dynamics problem, together with low-order spatial operator algorithms to reduce the cost of setting up the LCP. This results in a more than an order of magnitude reduction in computational cost. The size of the resulting LCP problem is independent of the number of links and generalized coordinates, and only depends on the number of contact nodes. We also describe extensions of the formulation to handle elastic and inelastic collision dynamics. The formulation is developed in progressive steps to clarify the trade offs and relationships among the methods. We use a multi-link pendulum numerical problem to quantitatively measure the performance improvements from the new OS formulation. A dual-arm robot system is used as a reference system to compare the LCP sizes for the different formulations discussed in this article.

The organization of this chapter is as follows. Section 5.2 describes the complementarity conditions associated with modeling a single unilateral contact constraint. Section 5.3 describes a system-level, multiple contacts *NMC LCP formulation* based on non-minimal coordinates. This formulation is easy to set up, but leads to a large LCP. Section 5.4 develops an alternative *MC formulation* that uses minimal coordinates. The reduction in the size of the LCP is however accompanied by an increase in the cost of setting up the LCP. Section 5.5 uses the MC LCP formulation to develop the *RMC formulation* that further reduces the size of the LCP problem, but once again at the cost of a further increase in the LCP setup cost. Section 5.6 finally develops the *OS* formulation that is based on an operational space approach. While this LCP's size is moderately larger than the RMC LCP, it is able to use low-order operational space algorithms to significantly reduce the LCP setup cost. Section 5.7 extends the OS formulation contact dynamics model to include elastic and inelastic collision dynamics. Section 5.8 focuses on computational issues, and describes the operational space computational algorithms to reduce the cost of setting up the OS LCP problem. The section also presents numerical simulation results to quantify the performance improvements for the OS formulation.

5.2 Unilateral Contact Constraints

Unilateral constraints are defined by inequality relationships of the form

$$\mathfrak{d}(\theta, t) \geq 0 \tag{5.1}$$

for some function $\mathfrak{d}$ of the configuration coordinates θ and time t. As an example, the non-penetration condition for rigid bodies can be stated as an inequality relationship requiring that the distance between the surfaces of rigid bodies be non-negative. $\mathfrak{d}(\theta, t)$ is generally referred to as the *distance* or *gap* function.

Contact occurs at the constraint boundary, i.e., when $\mathfrak{d}(\theta, t) = 0$. For bodies in contact, the surface normals at the contact point are parallel. The existence of contact is typically determined using geometric or collision detection techniques. For a pair of bodies A and B in contact, we use a convention where the ith contact normal $\hat{n}(i)$ is defined as pointing from body B towards body A, so that motion of A in the direction of the normal leads to a separation of the bodies. A unilateral constraint is said to be in an *active* state when

$$\mathfrak{d}(\theta, t) = \dot{\mathfrak{d}}(\theta, t) = \ddot{\mathfrak{d}}(\theta, t) = 0 \tag{5.2}$$

Thus, a unilateral constraint is active when there is contact, and the contact persists. Only active constraints generate constraint forces on the system. A constraint that is not active is said to be *inactive*. Contact *separation* occurs when the relative linear velocity of the contact points along the normal becomes positive and the contact points drift apart. A separating constraint is in the process of losing contact and transitioning to an inactive state. At the start of a separation event, we have

$$\mathfrak{d}(\theta, t) = \dot{\mathfrak{d}}(\theta, t) = 0 \quad \text{and} \quad \ddot{\mathfrak{d}}(\theta, t) > 0 \tag{5.3}$$

5.2.1 Contact Impulse for an Active Contact Constraint

We now describe contact force modeling using the approach in references [2, 3]. The 6-dimensional spatial impulse at the ith active contact constraint node has a zero angular moment component. Its non-zero linear impulse component $F_u(i) \in \mathcal{R}^3$ can be decomposed into normal and tangential (friction impulse) components

$$F_u(i) = F_n(i)\hat{n}(i) + F_t(i)\hat{t}(i) \tag{5.4}$$

where $\hat{t}(i)$ denotes a tangent plane vector for the ith contact pair. Assuming that the friction coefficient is $\mu(i)$, the magnitude of the tangential Coulomb frictional impulse is bounded by the magnitude of the normal component as follows:

$$\|F_t(i)\| \leq \mu(i) F_n(i) \tag{5.5}$$

When the bodies have non-zero relative linear velocities at the contact point, the contact is said to be a *sliding* contact. Otherwise, when the relative linear velocity is zero, the contact is said to be a *rolling* contact. During sliding, the tangential frictional impulse is in a direction opposing the linear velocity vector (which necessarily lies in the contact tangent plane) and Eq. 5.5 holds with an equality. Thus, the tangential

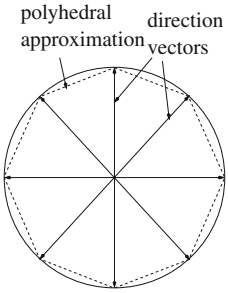


Fig. 5.2 Polyhedral approximation of the friction cone

friction impulse is on the boundary of the cone defined by Eq. 5.5 when sliding, and in the interior of the cone when rolling.

For the purpose of numerical computation, the friction cone at the ith contact is approximated by a friction polyhedron consisting of a finite number, n_f, of unit direction vectors $\hat{d}_j(i)$ in the tangent plane (see Fig. 5.2). It is assumed that for each direction vector, its opposite direction vector is also in the set. For notational simplicity, we assume that n_f is the same across all contact points. The ith contact tangential frictional impulse is expressed as the linear combination of these direction vectors as follows:

$$F_t(i)\hat{t}(i) = \sum_{j=1}^{n_f} \beta_j(i)\hat{d}_j(i) = D(i)\beta(i) \tag{5.6}$$

where

$$D(i) \triangleq \left[\hat{d}_1(i), \cdots, \hat{d}_{n_f}(i)\right] \in \mathcal{R}^{3\times n_f} \quad \text{and} \quad \beta(i) \triangleq \text{col}\left\{\beta_j(i)\right\}_{j=1}^{n_f} \in \mathcal{R}^{n_f}$$

Combining Eqs. 5.4 and 5.6 we have

$$F_u(i) = \underline{D}(i)\underline{\beta}(i), \quad \text{where} \quad \underline{\beta}(i) \triangleq \begin{bmatrix} F_n(i) \\ \beta(i) \end{bmatrix} \in \mathcal{R}^{n_f+1} \tag{5.7}$$

$$\text{and} \quad \underline{D}(i) \triangleq \left[\hat{n}(i),\ D(i)\right] \in \mathcal{R}^{3\times(n_f+1)}$$

During sliding, the $\beta_j(i)$ component is non-zero and equal to $\mu(i)F_n(i)$ for just the single direction j that corresponds to the closest direction opposing the (tangential) relative linear velocity. In other words, with $\sigma(i)$ denoting the magnitude of the contact relative linear velocity,

$$\beta_k(i) = \begin{cases} \mu(i)F_n(i)\mathbb{1}_{[k=j]} & \text{if} \quad \sigma(i) > 0 \\ 0 & \text{if} \quad \sigma(i) = 0 \end{cases} \tag{5.8}$$

In the above, $\mathbb{1}_{[<cond>]}$ denotes the indicator function whose value is 1 if the condition is true, and 0 otherwise.

5.2.2 Complementarity Relationship for a Unilateral Contact

We begin by defining complementarity conditions. Let $f(z) \in \mathcal{R}^n$ denote a function of a vector $z \in \mathcal{R}^n$, whose z_i elements have lower and upper bounds l_i and u_i respectively. The *complementarity condition*, $f(z) \perp z$, is said to hold when the following properties apply:

- $f_i(z) \geq 0 \quad \text{when} \quad z_i = l_i$
- $f_i(z) \leq 0 \quad \text{when} \quad z_i = u_i$
- $f_i(z) = 0 \quad \text{when} \quad l_i < z_i < u_i$

Typically the bounds are $l_i = 0$ and $u_i = \infty$, and we will assume this to be the case unless otherwise stated. For these bounds, the elements of $f(z)$ and z are non-negative, and the complementarity condition requires that for any i, only one of f_i or z_i can be positive. A complementarity condition is a *linear complementarity condition* when $f(z)$ has the form $\mathfrak{M}z + q \perp z$ for some matrix $\mathfrak{M}$ and vector q. Thus for an LCP

$$\mathfrak{M}z + q \perp z \tag{5.9}$$

We have a *mixed complementarity condition* when one or more of the rows of $f(z)$ are exactly equal to zero, i.e. the bounds for one or more of the rows are $l_i = -\infty$ and $u_i = \infty$. Such identically zero rows represent equality conditions while the rest represent are complementarity (inequality) conditions.

The sliding/rolling contact relationships described above can be rephrased as the following complementarity conditions[2]:

$$\hat{n}^*(i)v_u^+(i) \perp F_n(i) \quad (separation) \tag{5.10a}$$

$$\sigma(i)E(i) + D^*(i)v_u^+(i) \perp \beta(i) \quad (friction \quad force \quad direction) \tag{5.10b}$$

$$\mu(i)F_n(i) - E^*(i)\beta(i) \perp \sigma(i) \quad (friction \quad force \quad magnitude) \tag{5.10c}$$

where

$$E(i) \triangleq \text{col}\{1\}_{j=1}^{n_f} \in \mathcal{R}^{n_f} \tag{5.11}$$

and $v_u^+(i) \in \mathcal{R}^3$ denotes the relative linear velocity of the contact node on the first body A with respect to the contact node on the second body B. The component of

[2] For a vector/matrix A, the A^* notation denotes its vector/matrix transpose.

this relative linear velocity along the contact normal is, $\hat{n}^*(i)v_u^+(i)$. A positive value implies increasing separation between the bodies, while a negative value indicates that the bodies are approaching each other. Equation 5.10a states that this velocity component and the normal interaction impulse $F_n(i)$ cannot both be simultaneously positive. Thus the interaction impulse must be zero when the bodies are separating, and the impulse can be non-zero only if we have sustained contact. Equation 5.10b implies that the tangential friction impulse opposes the tangential relative linear velocity, while Eq. 5.10c states that the magnitude of the tangential impulse is on the friction cone boundary when the the tangential relative linear velocity is non-zero.

The complementarity conditions in Eq. (5.10a), (5.10b), (5.10c) enforce the no inter-penetration constraint at the velocity level instead of at the gap level. Hence they are valid only when the gap is zero, i.e., when contact exists [3]. Using Eqs. 5.7 and (5.10a), (5.10b), (5.10c) can be expressed more compactly as

$$\begin{aligned} \hat{E}(i)\sigma(i) + \underline{D}^*(i)v_u^+(i) \perp \underline{\beta}(i) \\ \bar{E}(i)\underline{\beta}(i) \perp \sigma(i) \end{aligned} \tag{5.12}$$

where

$$\begin{aligned} \hat{E}(i) &\triangleq \begin{bmatrix} 0 \\ E(i) \end{bmatrix} \in \mathcal{R}^{(n_f+1)} \\ \text{and} \quad \bar{E}(i) &\triangleq \begin{bmatrix} \mu(i), & -E^*(i) \end{bmatrix} \in \mathcal{R}^{1\times(n_f+1)} \end{aligned} \tag{5.13}$$

With n_u denoting the number of unilateral contact nodes, the component level complementarity conditions in Eq. 5.12 can be aggregated across all the contact constraints and expressed at the system level as:

$$\hat{E}\sigma + \underline{D}^* v_u^+ \perp \underline{\beta} \quad \text{and} \quad \bar{E}\underline{\beta} \perp \sigma \tag{5.14}$$

where

$$\begin{aligned} \underline{\beta} &\triangleq \text{col}\left\{\underline{\beta}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u(n_f+1)} \\ \sigma &\triangleq \text{col}\left\{\sigma(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u} \\ \underline{D} &\triangleq \text{diag}\left\{D(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u\times n_u(n_f+1)} \\ \hat{E} &\triangleq \text{diag}\left\{\hat{E}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u(n_f+1)\times n_u} \\ \bar{E} &\triangleq \text{diag}\left\{\bar{E}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u\times n_u(n_f+1)} \\ v_u^+ &\triangleq \text{col}\left\{v_u^+(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u} \end{aligned} \tag{5.15}$$

Also, Eq. 5.7 can be restated at the system level as

$$F_u = \underline{D}\underline{\beta} \quad \text{where} \quad F_u \triangleq \text{col}\,\{F_u(i)\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u} \tag{5.16}$$

5.3 Non-minimal Coordinates LCP Formulation

In this section we derive the commonly used non-minimal coordinate LCP formulation for contact dynamics based on the approach in [3]. We refer to this formulation as the *non-minimal coordinates* (*NMC*) formulation.

Contact and collision dynamics models build upon smooth dynamics models. The smooth dynamics model used by the NMC method treats all the links in the system as independent bodies, and all coupling hinges as explicit bilateral constraints as illustrated in Fig. 5.3. Such a smooth dynamics model utilizes non-minimal coordinates and is also referred to as a *fully augmented (FA)* model [13].

Let n denote the number of links in the system, and $\mathcal{N}$ the number of system degrees of freedom in the absence of bilateral constraints. For the FA model $\mathcal{N} = 6n$. Let n_b denote the dimension of the bilateral constraints arising from inter-link hinges and loop closure constraints on the system. With x denoting the vector of positional and attitude coordinates for the links, let $\mathcal{V} \in \mathcal{R}^{6n}$ denote the stacked vector of spatial velocities of all the links. Then there exists a $G_b(x,t) \in \mathcal{R}^{n_b \times 6n}$ matrix and a $\mathfrak{U}(t) \in \mathcal{R}^{n_b}$ vector that defines the following velocity domain constraint equation for the bilateral constraints on the system:

$$G_b(x,t)\mathcal{V} = \mathfrak{U}(t) \tag{5.17}$$

We assume that $G_b(x,t)$ is a *full-rank* matrix. Observe that Eq. 5.17 is linear in $\mathcal{V}$. The bilateral constraints effectively reduce the independent degrees of freedom for the system from $\mathcal{N}$ to $(\mathcal{N} - n_b)$. The bilateral constraints are accounted for via *Lagrange multipliers*, $\lambda \in \mathcal{R}^{n_b}$ to yield the following smooth equations of motion for the system

$$\begin{aligned} \mathcal{M}\alpha - G_b^*(x,t)\lambda &= C(x,\mathcal{V}) \\ G_b(x,t)\mathcal{V} &= \mathfrak{U}(t) \end{aligned} \tag{5.18}$$

where $\alpha \in \mathcal{R}^{6n}$ denotes the spatial acceleration of the bodies. $\mathcal{M} \in \mathcal{R}^{6n \times 6n}$ is a block diagonal matrix with the 6×6 spatial inertias of each of the links along the diagonal. $C \in \mathcal{R}^{6n}$ is a vector of the velocity dependent Coriolis and external forces on the system. The $-G_b^*(x,t)\lambda$ term in the first equation represents the constraint forces from the bilateral constraints. Differentiating the Eq. 5.17 constraint equation, Eq. 5.18 can be rearranged into the following descriptor form:

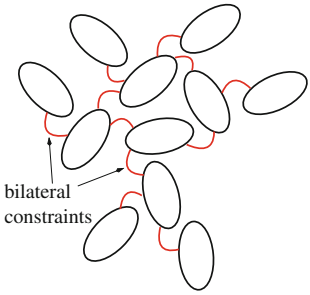


Fig. 5.3 Fully augmented model with hinges modeled as constraints

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \alpha \\ \lambda \end{bmatrix} = \begin{bmatrix} C \\ \bar{\mathfrak{U}} \end{bmatrix} \quad \text{where} \quad \bar{\mathfrak{U}} \triangleq \dot{\mathfrak{U}} - \dot{G}_b \mathcal{V} \in \mathcal{R}^{n_b} \tag{5.19}$$

An attractive feature of these smooth equations of motion is that the $\mathcal{M}$ matrix is block diagonal and constant. Using the following discrete time Euler step approximation over a Δ_t time interval,[3]

$$\mathcal{V}^+ - \mathcal{V}^- = \alpha \Delta_t \quad \text{and} \quad p_b \triangleq \lambda \Delta_t \in \mathcal{R}^{n_b} \tag{5.20}$$

the differential form of the equations of motion in Eq. 5.19 can be transformed into the following discretized version that maps the p_b impulse stacked vector at the bilateral constraint nodes into the resulting change in body spatial velocities.

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \end{bmatrix} = \begin{bmatrix} C\Delta_t \\ \bar{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{5.21}$$

5.3.1 Including Contact Impulses

The stacked vector of relative linear velocities across the contact nodes is denoted $v_u \in \mathcal{R}^{3n_u}$. It is related to the stacked vector of body spatial velocities $\mathcal{V}$ via the following relationship

$$v_u = G_u \mathcal{V} \tag{5.22}$$

where the $G_u \in \mathcal{R}^{3n_u \times 6n}$ matrix contains one block-row per contact node-pair, with each row mapping the spatial velocities for a node pair into the relative linear velocity across the contact. The G_u matrix also relates the F_u equal and opposite impulses at the contact node-pairs to the corresponding spatial impulses on the bodies, $p_u \in \mathcal{R}^{6n}$ via the following dual mapping

[3] The $-$ and $+$ superscripts denote the respective value of a quantity just before and after the application of an impulse.

$$p_u = G_u^* F_u \tag{5.23}$$

The p_u contact impulses can be included in the Eq. 5.21 smooth equations of motion by adding p_u to the $C\Delta_t$ term to obtain

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \end{bmatrix} = \begin{bmatrix} C\Delta_t + p_u \\ \bar{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{5.24}$$

5.3.2 Assembling the System LCP

We now set up an LCP to help solve the equations of motion and the unknown constraint forces. From Eqs. 5.16 to 5.23 we have

$$F_u = \underline{D}\underline{\beta} \quad \Rightarrow \quad p_u = G_u^* \underline{D}\underline{\beta} \tag{5.25}$$

Thus Eq. 5.24 can be recast as

$$\begin{pmatrix} \mathcal{M} & -G_b^* & -G_u^* \underline{D} \\ G_b & 0 & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \\ \underline{\beta} \end{bmatrix} = \begin{bmatrix} C\Delta_t \\ \bar{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{5.26}$$

Combining this with the complementarity conditions in Eq. 5.14 leads to the following NMC formulation of the LCP in Eq. 5.9:

$$\mathfrak{M} \triangleq \left(\begin{array}{ccc|c} \mathcal{M} & -G_b^* & -G_u^* \underline{D} & 0 \\ G_b & 0 & 0 & 0 \\ \hline \underline{D}^* G_u & 0 & 0 & \hat{E} \\ 0 & 0 & \bar{E} & 0 \end{array}\right), \quad z \triangleq \begin{bmatrix} \mathcal{V}^+ \\ p_b \\ \hline \underline{\beta} \\ \underline{\sigma} \end{bmatrix}, \quad q \triangleq \begin{bmatrix} -\mathcal{M}\mathcal{V}^- - C\Delta_t \\ -G_b\mathcal{V}^- - \bar{\mathfrak{U}}\Delta_t \\ \hline 0 \\ 0 \end{bmatrix} \tag{5.27}$$

This is a mixed LCP problem, where the first two rows are equality conditions, while the lower two rows are complementarity conditions. This NMC LCP formulation is essentially the one described in [3]. It makes use of non-minimal coordinates for the articulated system and is of size $(6n + n_b + n_u(n_f + 2))$. The constant and block-diagonal structure of $\mathcal{M}$ results in $\mathfrak{M}$ having a simple and highly sparse structure. The complexity of assembling $\mathfrak{M}$ and q for the LCP is just $O(n)$. Reference [3] derives sufficient conditions for the existence of a solution for the LCP problem.

The solution of the Eq. 5.27 LCP provides new $\mathcal{V}^+$ velocity coordinates which can be numerically integrated to propagate the x configuration coordinates. The solution values of $\underline{\beta}$ indicate which contacts are active or inactive, while the values of $\underline{\sigma}$ define the rolling or sliding state of each of the active contacts. Thus an LCP solution with $F_u(i)$ positive indicates that the ith contact is *active*. Furthermore, $\sigma(i) = 0$ implies

that the *i*th contact is a *rolling contact* while a positive value implies that it is a *sliding* contact.

In the NMC formulation, most of the computational effort involves solving the LCP, while the cost of setting up the LCP is relative low. The main disadvantage of this formulation is the large size of the LCP and the consequent large cost for solving it. Moreover, the use of non-minimal coordinates mandates the additional use of constraint error stabilization schemes to avoid the build up of constraint violation errors for the bilateral constraints.

We will use the dual-arm robot in Fig. 5.1 to track and compare the LCP size for this formulation and the ones to follow. This dual-arm platform has a 4 link sensor head, a pair of 7 link arms, with each arm having a 3 finger hand for an overall system with 26 links and 26 degrees of freedom. It has no loop closure bilateral constraints. Thus $n = 26, \mathcal{N} = 6n = 156$, and $n_b = 5n = 130$. For this exercise we assume that $n_f = 4$, and that there are 4 contact constraints. With these parameters, the size of the NMC LCP is 310 for the dual-arm system. The statistics for the NMC scheme are also summarized in the first column of Table 5.1 in Sect. 5.6.

5.4 Minimal Coordinate LCP Formulation

In contrast with the NMC formulation, in the *minimal coordinates* (*MC*) formulation, inter-link hinges are not modeled as bilateral constraints. Instead, minimal hinge coordinates are used to parameterize the permissible hinge motion. In doing so, the number of coordinates associated with the hinge match the number of degrees of freedom for the hinge. This approach is used for all the hinges in a spanning tree for the system graph, and bilateral constraints are used only for additional loop closures that may be present in the system topology as illustrated in Fig. 5.4.

Except for the switch from non-minimal to minimal coordinates, the development of the MC formulation largely parallels that for the NMC formulation. Hence wherever possible, we reuse the earlier notation, with the understanding that the meaning of each symbol depends on the formulation context. Thus once again, we use $\mathcal{N}$ to denote the number of degrees of freedom for the tree sub-system. With $\theta \in \mathcal{R}^{\mathcal{N}}$ denoting the vector of hinge coordinates, the minimal coordinates equations of motion for the smooth dynamics of just the tree-topology sub-system can be expressed as

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) = \mathcal{T} \tag{5.28}$$

where the configuration dependent matrix $\mathcal{M}(\theta) \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$ is the *mass matrix* of the system, $\mathcal{C}(\theta, \dot{\theta}) \in \mathcal{R}^{\mathcal{N}}$ denotes the velocity dependent Coriolis and gyroscopic forces vector, and $\mathcal{T} \in \mathcal{R}^{\mathcal{N}}$ denotes the applied generalized forces. The mass matrix is symmetric and positive-definite for tree-topology systems. The configuration dependency and dense structure of $\mathcal{M}$ makes it clearly more complex than the sparse structure and constant value of the $\mathcal{M}$ mass matrix in the NMC formulation. On the other

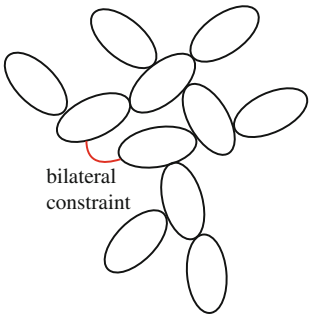


Fig. 5.4 Tree augmented model with only loop closures modeled as bilateral constraints

hand, for the dual-arm robot system in Fig. 5.1, $\mathcal{M}$ is a compact 26-dimensional square matrix compared with the 156-dimensional square matrix $\mathcal{M}$.

Let n_b denote the dimension of the bilateral constraints on the system arising from loop closures in the system. Since n_b applies only to loop bilateral constraints, it is much smaller than n_b in the NMC formulation. There exists a $G_b(\theta, t) \in \mathcal{R}^{n_b \times \mathcal{N}}$ matrix and a $\mathfrak{U}(t) \in \mathcal{R}^{n_b}$ vector that defines the velocity domain loop closure constraint equation as follows:

$$G_b(\theta, t)\dot{\theta} = \mathfrak{U}(t) \tag{5.29}$$

Once again we assume that $G_b(\theta, t)$ is a *full-rank* matrix.

The smooth dynamics of closed-chain systems can be obtained by modifying the tree system dynamics in Eq. 5.28 to include the effect of the bilateral constraints via *Lagrange multipliers*, $\lambda \in \mathcal{R}^{n_b}$, as follows

$$\begin{aligned} \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) - G_b^*(\theta, t)\lambda &= \mathcal{T} \\ G_b(\theta, t)\dot{\theta} &= \mathfrak{U}(t) \end{aligned} \tag{5.30}$$

By differentiating the bilateral constraint equation (Eq. 5.29), and including in the average force from the $p_u \in \mathcal{R}^{\mathcal{N}}$ contact impulse, Eq. 5.30 can be rearranged into the following descriptor form:

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} + p_u/\Delta_t \\ \bar{\mathfrak{U}} \end{bmatrix} \quad \text{where} \quad \bar{\mathfrak{U}} \triangleq \dot{\mathfrak{U}}(t) - \dot{G}_b\dot{\theta} \in \mathcal{R}^{n_c} \tag{5.31}$$

Using the discrete Euler step approximation

$$\dot{\theta}^+ - \dot{\theta}^- = \ddot{\theta}\Delta_t \tag{5.32}$$

the discretized version of Eq. 5.31 takes the form

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \dot{\theta}^+ - \dot{\theta}^- \\ p_b \end{bmatrix} = \begin{bmatrix} (\mathcal{T} - \mathcal{C})\Delta_t + p_u \\ \bar{\mathfrak{U}}\Delta_t \end{bmatrix} \quad \text{with} \quad p_b \triangleq \lambda \Delta_t \tag{5.33}$$

With $G_u \in \mathcal{R}^{3n_u \times \mathcal{N}}$ such that

$$v_u = G_u \dot{\theta} \tag{5.34}$$

the dual expression for the contact spatial impulses is given by

$$p_u = G_u^* F_u \overset{5.16}{=} G_u^* \underline{D} \underline{\beta} \tag{5.35}$$

Combining the complementarity conditions in Eqs. 5.14 with 5.33 leads to the MC formulation version of the Eq. 5.9 LCP with

$$\mathfrak{M} \triangleq \left(\begin{array}{cc|cc} \mathcal{M} & -G_b^* & -G_u^* \underline{D} & 0 \\ G_b & 0 & 0 & 0 \\ \hline \underline{D}^* G_u & 0 & 0 & \hat{E} \\ 0 & 0 & \bar{E} & 0 \end{array}\right) \tag{5.36}$$

$$\text{and} \quad z \triangleq \left[\begin{array}{c} \dot{\theta}^+ \\ \hline p_b \\ \hline \underline{\beta} \\ \underline{\sigma} \end{array}\right], \qquad q \triangleq \left[\begin{array}{c} -\mathcal{M}\dot{\theta}^- - (\mathcal{T} - \mathcal{C})\Delta_t \\ -G_b \dot{\theta}^- - \bar{\mathfrak{U}}\Delta_t \\ \hline 0 \\ 0 \end{array}\right]$$

This is a mixed LCP with the top two rows correspond to equality conditions while the lower two are complementarity conditions. Its structure is very similar to the NMC formulation LCP in Eq. 5.27 and differs primarily in the use of minimal coordinates. The size of the MC LCP is $(\mathcal{N} + n_b + n_u(n_f + 2))$. Unlike the NMC formulation, this dimension does not depend on the number of links n. Since $\mathcal{N}$ is much smaller when using minimal coordinates, the MC LCP size is much smaller than the NMC LCP size. For the dual arm robot in Fig. 5.1, the dimension of the MC LCP is just 50 compared with 310 for the NCP formulation.

On the other hand, evaluating $\mathfrak{M}$ for the MC LCP requires the configuration dependent and dense $\mathcal{M}$ mass matrix. While the composite rigid body inertia algorithm provides an efficient way to compute $\mathcal{M}$ [10], the computational cost scales as $O(\mathcal{N}^2)$. Thus the decrease in the LCP size and solution cost for the MC formulation is traded off for an increase in the cost of setting up the LCP. The computational complexity for the MC formulation is summarized in Table 5.1. The solution of the MC LCP yields the new $\dot{\theta}^+$ generalized velocity value which can be integrated to propagate the θ configuration coordinates. As in the case of the NMC formulation, the bulk of the computational effort in the MC formulation is in setting up and solving the LCP problem.

5.5 Reduced Minimal Coordinate LCP Formulation

Continuing with the minimal coordinate approach, we now take further steps to reduce the size of the LCP problem. The matrix on the left of Eq. 5.31 can be inverted to yield the following solution for $\ddot{\theta}$:

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + p_u/\Delta_t\right] \tag{5.37a}$$

$$\lambda = \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1}\left(-G_b \ddot{\theta}_f + \bar{\mathfrak{U}}\right) \tag{5.37b}$$

$$\ddot{\theta} = \ddot{\theta}_f + \mathcal{M}^{-1} G_b^* \lambda \overset{5.37b}{=} \left[I - \mathcal{M}^{-1} G_b^* \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1} G_b\right] \ddot{\theta}_f + \mathcal{M}^{-1} G_b^* \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1} \bar{\mathfrak{U}} \tag{5.37c}$$

Using Eq. 5.32, we obtain

$$\begin{aligned} \dot{\theta}^+ &\overset{5.32}{=} \dot{\theta}^- + \ddot{\theta}\Delta_t \\ &\overset{5.37c}{=} \dot{\theta}^- + \left[I - \mathcal{M}^{-1} G_b^* \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1} G_b\right] \Delta_t \ddot{\theta}_f + \mathcal{M}^{-1} G_b^* \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1} \Delta_t \bar{\mathfrak{U}} \\ &\overset{5.37a}{=} Y p_u + X \end{aligned} \tag{5.38}$$

where

$$Y \triangleq \mathcal{M}^{-1} - \mathcal{M}^{-1} G_b^* (G_b \mathcal{M}^{-1} G_b^*)^{-1} G_b \mathcal{M}^{-1} \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \tag{5.39}$$

$$\text{and} \quad X \triangleq \dot{\theta}^- + Y(\mathcal{T} - \mathcal{C})\Delta_t + \mathcal{M}^{-1} G_b^* \left[G_b \mathcal{M}^{-1} G_b^*\right]^{-1} \bar{\mathfrak{U}} \Delta_t \in \mathcal{R}^{\mathcal{N}}$$

Thus

$$\underline{D}^* v_u^+ \overset{5.34}{=} \underline{D}^* G_u \dot{\theta}^+ \overset{5.35,5.38}{=} \underline{D}^* G_u Y G_u^* \underline{D}\, \underline{\beta} + \underline{D}^* G_u X \tag{5.40}$$

Using this allows us to eliminate $\dot{\theta}^+$ and p_b from the MC LCP formulation in Eq. 5.36 to obtain the following *Reduced Minimal Coordinate* (*RMC*) formulation LCP:

$$\mathfrak{M} \triangleq \begin{pmatrix} \underline{D}^* G_u Y G_u^* \underline{D} & \hat{E} \\ \bar{E} & 0 \end{pmatrix}, \quad z \triangleq \begin{bmatrix} \underline{\beta} \\ \sigma \end{bmatrix}, \quad q \triangleq \begin{bmatrix} \underline{D}^* G_u X \\ 0 \end{bmatrix} \tag{5.41}$$

Since there are no equality conditions, this is a standard rather than a mixed LCP. The size of this RMC LCP is $n_u(n_f + 2)$. It is notable that the size of the LCP does not depend on the number of links n, the number of degrees of freedom $\mathcal{N}$, nor the n_b dimension of the bilateral constraints. It only depends on the number of contact constraint nodes. Thus the dimension of this LCP is even smaller than that for the MC formulation. For the dual arm robot system, the dimension of the LCP is 24. On the other hand, computing $\mathfrak{M}$ for the RMC LCP requires the Y matrix in Eq. 5.39, which requires the configuration dependent $\mathcal{M}^{-1}$ matrix and several expensive matrix/matrix products. These computations are of $O(\mathcal{N}^3)$ computational complexity. Once again, while the RMC formulation successfully reduces the LCP size and consequently its solution cost, this reduction is accompanied by a significant increase in the cost of setting up the LCP. The computational complexity for the RMC formulation is summarized in Table 5.1.

In contrast with the NMC and MC formulations, the solution of the RMC LCP does not by itself yield the new system velocity or state. Instead the following sequence of steps is needed to obtain the new state values:

1. Assemble and solve the RMC LCP in Eq. 5.41 to obtain $\underline{\beta}$ and σ. Use $\underline{\beta}$ in Eq. 5.35 to obtain the p_u contact impulse vector.
2. Use p_u in Eq. 5.38 to compute the new $\dot{\theta}^+$ system velocity. This can be integrated to obtain the new system configuration coordinates θ.

Thus, the RMC LCP by itself does not do all the work, and the additional step (2) is needed to complete the computation of the new $\dot{\theta}^+$ system velocity coordinates.

The formulation developed by Trinkle [2] is a hybrid combination of the NMC and RMC formulations. Trinkle's setup allows the use of general coordinates for describing the smooth equations of motion. However, instead of eliminating the hinge bilateral constraints by using minimal hinge coordinates a pair of symmetric (positive and negative) complementarity conditions are added to enforce the equality condition for each hinge constraint. This inflates the size of the LCP much like the NMC approach. However, Trinkle;s approach is similar to the RMC in eliminating the velocity coordinates and the loop closure bilateral constraint Lagrange multipliers from the LCP problem to obtain an LCP similar in form to Eq. 5.41.

5.6 Operational Space LCP Formulation

So far we have found that the reductions in LCP size have the side-effect of increasing the LCP setup cost. In this section we look into reducing such setup cost using low-order structure-based dynamics algorithms. Using

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}[\mathcal{T} - \mathcal{C}] \quad \text{and} \quad \delta_b^f \triangleq G_b \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}) - \bar{\mathfrak{A}} = G_b \ddot{\theta}_f - \bar{\mathfrak{A}} \tag{5.42}$$

in Eq. 5.31, we obtain

$$
\begin{aligned}
0 &\overset{5.29}{=} G_b\ddot{\theta} - \bar{\mathfrak{U}} \\
&\overset{5.31}{=} G_b\mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + G_b^*\lambda + p_u/\Delta_t\right] - \bar{\mathfrak{U}} \\
&= G_b\mathcal{M}^{-1}G_b^*\lambda + G_b\mathcal{M}^{-1}p_u/\Delta_t + \delta_b^f \\
&\overset{5.35}{=} G_b\mathcal{M}^{-1}G_b^*\lambda + G_b\mathcal{M}^{-1}G_u^*\underline{D}\underline{\beta}/\Delta_t + \delta_b^f
\end{aligned}
\tag{5.43}
$$

The above expression characterizes the equality condition on the dynamics from the bilateral constraints. Observe that $\ddot{\theta}_f$ represents the generalized acceleration that would occur in the absence of the bilateral and contact constraints, and can be regarded as the *free* generalized acceleration for the system. For this hypothetical free system, δ_b^f represents the time derivative of the velocity residual $G_b\dot{\theta} - \mathfrak{U}(t)$ for the bilateral constraints. For $\dot{\theta}$ consistent with the constraints, clearly this velocity residual is instantaneously zero, but it has the δ_b^f non-zero time derivative were the system dynamics to evolve according to the free dynamics. In reality, the system dynamics is constrained and this velocity residual and its time derivative remain zero.

The relative linear acceleration of the contact nodes is obtained by differentiating Eq. 5.34 to obtain

$$
\dot{v}_u = G_u\ddot{\theta} + \dot{G}_u\dot{\theta} \overset{5.31}{=} G_u\mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + G_b^*\lambda + p_u/\Delta_t\right] + \dot{G}_u\dot{\theta} \tag{5.44}
$$

With

$$
\delta_u^f \overset{\triangle}{=} G_u\mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}) + \dot{G}_u\dot{\theta} = G_u\ddot{\theta}_f + \dot{G}_u\dot{\theta} \tag{5.45}
$$

the discretized approximation $(v_u^+ - v_u^-) = \dot{v}_u\Delta_t$ of Eq. 5.44 leads to

$$
v_u^+ \overset{5.35,\,5.44}{=} G_u\mathcal{M}^{-1}G_b^*\lambda\Delta_t + G_u\mathcal{M}^{-1}G_u^*\underline{D}\,\underline{\beta} + v_u^- + \delta_u^f\Delta_t \tag{5.46}
$$

Physically, δ_u^f is the time derivative of the contact relative velocity v_u^- were the system to evolve in accordance with the free dynamics, i.e. in the absence of the bilateral and contact constraints. Combining the complementarity conditions in Eq. 5.14 with Eqs. 5.43 and 5.46 yields the following mixed LCP for the system:

$$
\mathfrak{M} \overset{\triangle}{=} \left(\begin{array}{cc|c}
G_b\mathcal{M}^{-1}G_b^* & G_b\mathcal{M}^{-1}G_u^*\underline{D} & 0 \\
\underline{D}^*G_u\mathcal{M}^{-1}G_b^* & \underline{D}^*G_u\mathcal{M}^{-1}G_u^*\underline{D} & \hat{E} \\
\hline
0 & \bar{E} & 0
\end{array}\right) \tag{5.47}
$$

$$\text{and} \quad z \triangleq \begin{bmatrix} p_b \\ \underline{\beta} \\ \sigma \end{bmatrix}, \qquad q \triangleq \begin{bmatrix} \delta_b^f \Delta_t \\ \underline{D}^*(v_u^- + \delta_u^f \Delta_t) \\ 0 \end{bmatrix}$$

This $\mathfrak{M}$ matrix still requires the configuration dependent $\mathcal{M}^{-1}$ matrix whose evaluation if of $O(\mathcal{N}^3)$ computational complexity. We next look more closely at the structure of the G_u and G_b matrices.

The unilateral and bilateral constraints are associated with nodes on the bodies. Let us denote the number of this overall set of nodes involved in the unilateral and bilateral constraints as n_c. Denoting the spatial velocities of these nodes by the stacked vector $\mathcal{V}_c \in \mathcal{R}^{6n_c}$, there exist matrices $\mathcal{Q}_u \in \mathcal{R}^{3n_u \times 6n_c}$ and $\mathcal{Q}_b \in \mathcal{R}^{n_b \times 6n_c}$ such that the unilateral and bilateral velocity constraint equations can be expressed as[4]

$$v_u = \mathcal{Q}_u \mathcal{V}_c \quad \text{and} \quad \mathcal{Q}_b \mathcal{V}_c = \mathfrak{U} \tag{5.48}$$

Let $\mathcal{J} \in \mathcal{R}^{6n_c \times \mathcal{N}}$ denote the Jacobian for the constraint nodes, so that

$$\mathcal{V}_c = \mathcal{J}\dot{\theta} \tag{5.49}$$

It follows from Eqs. 5.29, 5.34, 5.48 and 5.49 that G_u and G_b have the following form:

$$G_u = \mathcal{Q}_u \mathcal{J} \quad \text{and} \quad G_b = \mathcal{Q}_b \mathcal{J} \tag{5.50}$$

With

$$\underline{\Lambda} \triangleq \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^* \in \mathcal{R}^{6n_c \times 6n_c} \tag{5.51}$$

we can use Eq. 5.50 to re-express $\mathfrak{M}$ in Eq. 5.47 as

$$\mathfrak{M} = \begin{pmatrix} \mathcal{Q}_b \underline{\Lambda} \mathcal{Q}_b^* & \mathcal{Q}_b \underline{\Lambda} \mathcal{Q}_u^* \underline{D} & 0 \\ \underline{D}^* \mathcal{Q}_u \underline{\Lambda} \mathcal{Q}_b^* & \underline{D}^* \mathcal{Q}_u \underline{\Lambda} \mathcal{Q}_u^* \underline{D} & \hat{E} \\ 0 & \bar{E} & 0 \end{pmatrix} = \left(\begin{array}{cc|c} \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \begin{bmatrix} \mathcal{Q}_b^*, & \mathcal{Q}_u^* \underline{D} \end{bmatrix} & & \begin{matrix} 0 \\ \hat{E} \end{matrix} \\ \hline 0 & \bar{E} & 0 \end{array} \right) \tag{5.52}$$

The $\underline{\Lambda} = \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^*$ matrix definition in Eq. 5.51 is precisely the mathematical expression for the inverse of the *operational space inertia* matrix that is used in the operational space approach for robot manipulation and control [14, 15]. Based on this structural similarity, we borrow and extend the operational space terminology to our current context with the constraint nodes forming the operational space nodes. Also, borrowing terminology, we refer to $\underline{\Lambda}$ as the *operational space compliance matrix* (*OSCM*) matrix. The invertibility of $\underline{\Lambda}$ does not depend on $\mathcal{J}$ being invertible—only

[4] $\mathcal{Q}_u$ has the same structure as would the $\mathcal{Q}_b$ constraint mapping matrix corresponding to bilateral constraints involving three degree of freedom spherical hinges.

that $\mathcal{J}$ have full row-rank. When it exists, the inverse of $\underline{\Lambda}$ is referred to as the *operational space inertia*. The properties of the OSCM are discussed in detail in [12].

The property of the $\underline{\Lambda}$ matrix that is of importance for us is the availability of algorithms of $O(\mathcal{N}) + O(n_c^2)$ computational complexity for evaluating $\underline{\Lambda}$ [12, 16]. The low-order of these algorithms is remarkable given the presence of $\mathcal{M}^{-1}$ in the expression for $\underline{\Lambda}$, since evaluating $\mathcal{M}$ and $\mathcal{M}^{-1}$ individually require $O(\mathcal{N}^2)$ and $O(\mathcal{N}^3)$ computations respectively. This algorithm reduces the complexity of evaluating $\mathfrak{M}$ in Eq. 5.52 from $O(\mathcal{N}^3)$ to the much smaller $O(\mathcal{N}) + O(n_c^2)$ computational complexity. The low complexity algorithm for evaluating $\underline{\Lambda}$ is based on an analytical transformation of Eq. 5.51, followed by a disjoint decomposition of the matrix into block diagonal, and upper and lower triangular components that can be computed recursively. A summary of this structure-based analysis and accompanying algorithms using spatial operator techniques is described in the appendix. An alternative sparsity based technique for evaluating $\underline{\Lambda}$ is described in reference [17].

Using Eq. 5.52 the Eq. 5.47 LCP can be re-expressed as the following *Operational Space (OS)* formulation LCP:

$$\mathfrak{M} \triangleq \left(\begin{array}{cc|c} \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \begin{bmatrix} \mathcal{Q}_b^*, & \mathcal{Q}_u^* \underline{D} \end{bmatrix} & & \begin{matrix} 0 \\ \hat{E} \end{matrix} \\ \hline 0 & \bar{E} & 0 \end{array}\right) \tag{5.53}$$

$$\text{and} \quad z \triangleq \begin{bmatrix} p_b \\ \beta \\ \hline \sigma \end{bmatrix}, \qquad q \triangleq \begin{bmatrix} \delta_b^f \Delta_t \\ \underline{D}^* (v_u^- + \delta_u^f \Delta_t) \\ \hline 0 \end{bmatrix}$$

This is a mixed LCP, with the first row corresponding to an equality condition while the bottom two rows correspond to complementarity conditions. The size of this LCP is $(n_b + n_u(n_f + 2))$. Like the RMC formulation, the size of this LCP does not depend on the number of links n or the number of degrees of freedom $\mathcal{N}$, but it does depend on the n_b dimension of the loop closure bilateral constraints. The dimension of the OS LCP is moderately larger than the RMC LCP but smaller than the MC LCP. Typically, $\dot{\mathcal{Q}}_u$, $\dot{\mathcal{Q}}_b$ and $\dot{\mathfrak{U}}$ are all zero leading to a simpler q in Eq. 5.53. For the dual arm robot system, the dimension of the OS LCP is 24.

Computing $\mathfrak{M}$ for the OS LCP requires the configuration dependent $\underline{\Lambda}$ matrix Eq. 5.53 whose evaluation is of $O(\mathcal{N}) + O(n_c^2)$ computational complexity which is much smaller than the $O(\mathcal{N}^3)$ complexity for evaluating $\mathfrak{M}$ for the RMC method. Thus in comparison with the RMC formulation, while the OS formulation increases the size of the LCP by a modest n_b, it drastically reduces the LCP setup complexity. The result is a significant reduction in the overall complexity of the contact dynamics computations for the OS formulation.

Like the RMC formulations, the solution of the LCP does not by itself yield the new system velocity or state. Instead the following sequence of steps is needed to obtain the new state values:

Table 5.1 A comparison of the features of the different NMC, MC, RMC and OS formulations for contact and collision dynamics

Property	LCP Formulation			
	NMC	MC	RMC	OS
Coordinates type	Non-minimal	Minimal	Minimal	Minimal
LCP assembly complexity	O(n)	$O(\mathcal{N}^2)$	$O(\mathcal{N}^3)$	$O(\mathcal{N}) + O(n_c^2)$
LCP dimension	$6n + n_b + n_u(n_f + 2)$	$\mathcal{N} + n_b + n_u(n_f + 2)$	$n_u(n_f + 2)$	$n_b + n_u(n_f + 2)$
Dual-arm LCP dimension	310	50	24	24
Ancillary dynamics steps	None	None	Evaluate p_u and $\dot{\theta}^+$	Evaluate p_u and $\dot{\theta}^+$

The LCP dimension size is for the reference dual-arm robot problem, while the LCP assembly complexity highlights just the major contributors

1. Assemble and solve the OS LCP in Eq. 5.53 to obtain p_b, $\underline{\beta}$ and σ. Use $\underline{\beta}$ in 5.35 to obtain the p_u contact impulse vector.
2. Use $\lambda = p_b/\Delta_t$ and p_u in Eq. 5.31 to obtain and integrate the $\ddot{\theta}$ generalized acceleration over the Δ_t time interval using *any* smooth integrator to obtain the new system state $(\theta, \dot{\theta})$.

Like the RMC formulation, the LCP by itself does not do all the work in the OS formulation, but instead the additional step (2) is needed to complete the computation of the new $\dot{\theta}^+$ system velocity coordinates.

The LCP formulation developed in references [8, 18] make use of the *divide and conquer algorithm (DCA)* [19] techniques and is a special case of the OS formulation. Our OS formulation is more general since it handles loop closure bilateral constraints, exploits operational space techniques to reduce computational complexity, and as described later, handles collision dynamics.

Table 5.1 summarizes the dimensions and computational complexity for all the formulations discussed so far. The trend across the NMC, MC and RMC formulations is that the reduction in the size of the LCP shifts costs to the LCP setup process. While the initial form of the OS formulation LCP in Eq. 5.47 also follows this trend, the restructured Eq. 5.53 LCP breaks the pattern by restructuring the LCP to take advantage of low-order, structure-based algorithms for the OSCM.

5.7 Collision Dynamics

In this section we develop extensions to the OS LCP formulation for handling the dynamics of collision events. During inelastic collisions some of the impact energy is lost. The *coefficient of restitution*, $\epsilon(i)$ defines the fraction that remains after a collision. The complementarity approach to modeling collisions breaks up the collision

event into a pair of instantaneous *compression* and *decompression* phases [3]. During the compression phase, the collision impulse is stored, and during decompression, a fraction of the collision impulse is recovered. We will make use of time discretized equations with impulses developed for contact dynamics, but with $\Delta_t = 0$ since collision events are assumed to be instantaneous.

5.7.1 Compression

At the ith contact undergoing collision, the compression phase is instantaneous and impulsively changes the relative linear contact velocity from $v_u^-(i)$ to a new $v_c^+(i)$ value with a non-negative normal component. The compression impulse is denoted $p_c(i)$. The mixed LCP problem for the compression phase is obtained by setting $\Delta_t = 0$ in Eq. 5.53 to obtain

$$w = \mathfrak{M} z + q \;\perp\; z \quad \text{with} \quad q \triangleq \left(\begin{bmatrix} 0 \\ \underline{D}^* v_u^- \\ 0 \end{bmatrix} \right) \tag{5.54}$$

The LCP solution is used to instantaneously (i.e. impulsively) propagate the state for the compression phase as follows:

$$\begin{aligned} p_c &= \mathcal{Q}_u^* \underline{D}\, \underline{\beta} + \mathcal{Q}_b^* p_b \\ \dot{\theta}^c &= \dot{\theta}^- + \mathcal{M}^{-1} \mathcal{J}^* p_c \\ v_c^+ &= \mathcal{J} \dot{\theta}^c \end{aligned} \tag{5.55}$$

5.7.2 Decompression

The decompression phase applies an additional impulse of magnitude $\epsilon(i)[0,\ \hat{n}^*(i) p_c(i)]$ for the ith contact along the normal from the impulse stored during the compression phase. The recovered ϑ decompression impulse is

$$\vartheta \triangleq \operatorname{col} \left\{ \left(\epsilon(i)[0,\ \hat{n}^*(i)] p_c(i) \right) \hat{n}(i) \right\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u} \tag{5.56}$$

The decompression LCP is obtained by updating Eqs. 5.43 and 5.44 to include the additional ϑ impulse. This leads to a decompression LCP problem that is the mixed LCP in Eq. 5.53 with $\Delta_t = 0$, the contact linear velocity v_u^- replaced with v_c^+, and an additional $\begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda}\, \mathcal{Q}_u^* \vartheta$ term for the recovered impulse included in the q LCP vector term. The resulting decompression phase LCP is

$$w = \mathfrak{M}z + q \perp z \quad \text{with} \quad q \triangleq \left(\begin{bmatrix} 0 \\ \underline{D}^* v_c^+ \end{bmatrix} + \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \\ 0 \end{bmatrix} \underline{\Lambda} \mathcal{Q}_u^* \vartheta \right) \tag{5.57}$$

The LCP solution for the decompression impulse can include additional contact impulse terms that ensure that the normal component of the relative linear velocity at the end of the decompression step remains non-negative. The LCP solution is used to instantaneously propagate the state for the decompression phase as follows:

$$\begin{aligned} p &= \mathcal{Q}_u^* \underline{D}\, \underline{\beta} + \mathcal{Q}_b^* \lambda + \mathcal{Q}_u^* \vartheta \\ \dot{\theta}^+ &= \dot{\theta}^c + \mathcal{M}^{-1} \mathcal{J}^* p \end{aligned} \tag{5.58}$$

when $\epsilon(i) = 0$, the collision is completely inelastic, and there is no decompression phase. However, in general, each collision event requires the solution of two LCP's in this approach.

5.8 Simulation Results

We use a simulation of a multi-link pendulum colliding with itself and the environment to quantitatively evaluate the performance of the OS formulation. This example also allows us to parameterically measure the performance improvement as a function of the problem dimension by varying the number of links in the pendulum. The environment consists of a floor and a wall located 4 m away. The multi-link pendulum consists of n identical 1 kg mass spherical bodies connected with pin hinges. The radius of the sphere is scaled based on the number of links to maintain a 12 m overall length of the pendulum. The pendulum base is located at a height of 10 m. The open source Bullet software [20] is used for collision detection, and the PATH software [21] for solving mixed complementarity problems. The simulation uses a time step of 1 ms, with a 0.5 coefficient of friction and a 0.5 coefficient of restitution to simulate inelastic collisions. The pendulum starts at an angle of $\pi/4$ radians with an initial angular velocity of 1 radian/s and a gravitational acceleration of 9.8 m/s^2.

As the pendulum swings from left to right, it collides with the ground, bounces off of the ground, and eventually collides with the wall on the right. In the course of the sequence, multiple links are at times in collision with the ground, the wall and with each other. Figure 5.5 contains a sequence of screen shots from such a simulation for a 12-link pendulum. We have simulated this contact and collision dynamics scenario using two different techniques. The first technique is the minimal coordinate OS formulation described in Sect. 5.6.

The second technique, that we refer to as the *NMC/OS formulation*, is a non-minimal coordinate variant of the OS formulation. Similar to the NMC method, each link is treated as an independent body, and the hinges are handled as bilateral

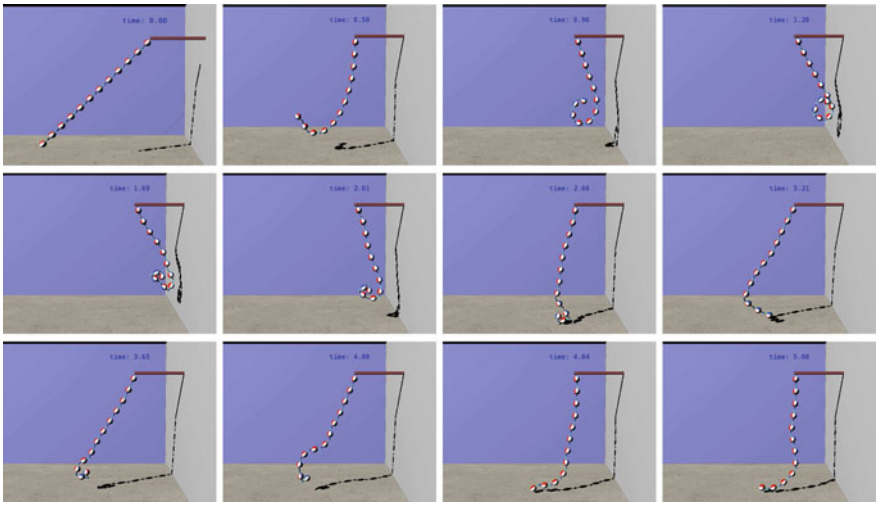


Fig. 5.5 Time series capture of swinging pendulum simulation with 12 links

constraints between the neighboring links with $n_b = 6n - \mathcal{N}$. The NMC/OS LCP has the same form as the OS LCP in Eq. 5.53, except that the OSCM is the non-minimal coordinate $\underline{\Lambda} = \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^*$, instead of Eq. 5.51. The NMC/OS $\underline{\Lambda}$ is a much larger matrix than for the OS formulation, but has a much simpler block diagonal structure. However, the NMC/OS LCP does not include system velocity coordinates $\mathcal{V}$ in z and thus is smaller than the NMC LCP.

Figure 5.6 shows example plots of the height and normal velocity of the last link of the 12-body pendulum from the two simulation methods. The simulation results from the two methods show good agreement through the first few collisions, with some divergence during the later phases. Reasons for the divergence include the widely differing choices of coordinates, and more importantly the The vertical spikes in the velocity plot are discontinuous jumps from collisions involving the pendulum bodies. The small trajectory differences in the plots decrease further when the time step size is reduced.

Table 5.2 compares the computational cost of the OS and the NMC/OS formulations for pendulums with the number of links varying between 3 and 30 links. The table also lists the LCP size for the OS, NMC/OS and the NMC formulations. The size of the LCP remains a constant value of 24 for the standard OS formulation even when the number of links and degrees of freedom in the system is increased. In contrast, the LCP size increases with the increase in the number of links and degrees of freedom for the NMC/OS and the NMC formulations. We also observe that the OS method is about 3.5 times faster for the 3 link pendulum case, and over 50 times faster for the 30 link pendulum when compared with the the NMC/OS method. The performance gap widens substantially as the number of links in the system is

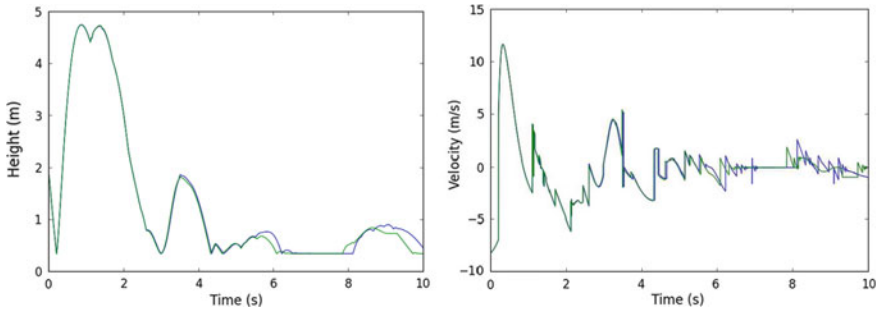


Fig. 5.6 Comparisons of the height and normal velocity of the last link using the OS (*red*) and NMC/OS (*blue*) formulation based simulations for a 12-body pendulum

Table 5.2 A comparison of the LCP size and computational time for the OS and NMC/OS formulations for the multi-link pendulum example with different number of links

Number of links	LCP size			Computation time (s)		
	OS	NMC/OS	NMC	OS	NMC/OS	Speed up
3	24	39	57	0.63	2.20	3.5
6	24	54	90	1.0	4.44	4.44
12	24	84	156	1.88	15.7	8.36
15	24	99	189	2.91	33.59	11.5
24	24	144	288	4.76	127.94	26.88
30	24	174	354	5.13	257.72	50.28

The LCP size assumes 4 contacts and $n_f = 4$. The three LCP size columns are for the OS, the NMC/OS and the NMC formulations. The speed up value is the ratio of the NMC/OS to the OS formulation simulation times

increased. The performance gap between the OS and NMC formulations will be even greater due to the even larger size of the NMC LCP.

5.9 Conclusions

In this article we have described a progressive series of formulations for solving the contact and collision dynamics of multi-link articulated systems with the goal of reducing computational cost. Along the way, we have clarified the relationships among the different approaches and those in the literature. Our strategy has been to use minimal coordinates and identify formulations that best exploit the available low-order structure based dynamics algorithms to reduce the overall computational cost.

The formulations studied here vary in the size of the LCP, the cost of setting up the LCP, and the ancillary dynamics steps needed to complete the dynamics solution. The generally observed trend is that the reduction in the LCP size shifts the computational

burden from solving the LCP problem, to the setting up of the LCP problem. The widely used NMC non-minimal coordinate formulation is the simplest and cheapest to set up, but also the most expensive to solve due to its large dimension. The RMC minimal coordinate approach on the other hand has the smallest LCP dimension, but one that is the most expensive to set up. In the RMC approach, the size of the LCP problem in is just $(n_u(n_f + 2) + n_b)$, which is independent of the number of links, the number of degrees of freedom and the dimension of the bilateral constraints on the system. In contrast, the size of the corresponding NMC LCP is larger by $6n - \mathcal{N}$. For a 6-link manipulator with 6 degrees of freedom, this amounts to an increase in dimension of 30.

The OS formulation also has the small LCP dimension property, with dimension exceeding that of the RMC LCP by just the (typically small) dimension of the loop closure bilateral constraints, n_b. The advantage of the OS formulation is that the LCP matrix can be expressed in terms of the operational space OSCM matrix for the constraint nodes. This form allows us to use low-order, structure-based computational algorithms available for the OSCM to significantly reduce the cost of setting up the OS LCP. Consequently the OS formulation has the lowest overall cost with a small LCP as well as low cost algorithms for setting up the LCP. Focusing on this option, we describe extensions of the contact dynamics formulation to handle elastic and inelastic collision dynamics. The OS formulation's use of minimal coordinates also results in the automatic enforcement of the inter-link hinge bilateral constraints and avoids the need for additional bilateral constraint error control schemes. The benchmark simulations using a pendulum system show a widening performance improvement using the OS formulation as the number of bodies is increased. For the 30 link pendulum system, the OS formulation is over 50 times faster than the NMC/OS approach. An area of future work is the extension of the OS formulation to use the more accurate nonlinear complementarity problem techniques, and time stepping schemes that are in development for increasing the robustness and accuracy of contact and collision non-smooth dynamics [22].

Acknowledgments The research described in this chapter was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration (©2014 California Institute of Technology. Government sponsorship acknowledged).

Appendix

The operational space for the multi-link system is defined by the configuration of the set of constraint nodes on the system. The key implementation and computational challenge for setting up the OS formulation LCP in Eq. 5.53 is the need for evaluating the $\underline{\Lambda}$ matrix. As seen in Eq. 5.51, $\underline{\Lambda}$ involves the configuration dependent matrix products of the Jacobian matrix and the mass matrix inverse. A direct evaluation of this expression requires $O(\mathcal{N}^3)$ computations. However references [12, 16, 23] have used spatial operators to develop simpler and recursive computational algorithms for

$\underline{\Lambda}$ that are of only $O(\mathcal{N})$ complexity. We briefly describe the underlying analysis and structure of this algorithm, and refer the reader to [12, 16, 23] for notation and derivation details.

Spatial Operator Factorization of $\mathcal{M}^{-1}$

We begin with the following key spatial operator based analytical results that provide explicit, closed-form expressions for the factorization and inversion of a tree mass matrix [12, 24]:

$$\begin{aligned} \mathcal{M} &= H\phi\mathcal{M}\phi^* H^* \\ \mathcal{M} &= [I + H\phi\mathcal{K}]\,\mathcal{D}\,[I + H\phi\mathcal{K}]^* \\ [I + H\phi\mathcal{K}]^{-1} &= [I - H\psi\mathcal{K}] \\ \mathcal{M}^{-1} &= [I - H\psi\mathcal{K}]^*\,\mathcal{D}^{-1}\,[I - H\psi\mathcal{K}] \end{aligned} \tag{5.59}$$

The first expression defines the Newton-Euler operator factorization of the mass matrix $\mathcal{M}$ in terms of the H hinge articulation, the ϕ rigid body propagation and the $\mathcal{M}$ link spatial inertia operators. While this factorization has non-square factors, the second expression describes an alternative factorization involving only square factors with block diagonal $\mathcal{D}$ and block lower-triangular $[I + H\phi\mathcal{K}]$ matrices. This factorization involves new spatial operators that are associated with the *articulated body* (*AB*) forward dynamics algorithm [11, 23] for the system. The next expression describes an analytical expression for the inverse of the $[I + H\phi\mathcal{K}]$ operator. Using this leads to the final analytical expression for the inverse of the mass matrix. These operator expressions hold generally for tree-topology systems irrespective of the number of bodies, the types of hinges, the specific topological structure, and even for non-rigid links [12].

The Ω Extended Operational Space Compliance Matrix

With $\mathcal{V} \in \mathcal{R}^{6n}$ denoting the stacked vector of link spatial velocities, its spatial operator expression is [12]

$$\mathcal{V} = \phi^* H^* \dot{\theta} \tag{5.60}$$

Bundling together the rigid body transformations for all nodes we define the $\mathcal{B} \in \mathcal{R}^{6n \times 6n_c}$ *pick-off* matrix such that the stacked vector of node spatial velocities $\mathcal{V}_c$ can be expressed as

$$\mathcal{V}_c = \mathcal{B}^* \mathcal{V} \overset{5.60}{=} \mathcal{B}^* \phi^* H^* \dot{\theta} \quad \Rightarrow \quad \mathcal{J} \overset{5.49}{=} \mathcal{B}^* \phi^* H^* \tag{5.61}$$

This is the spatial operator expression for the $\mathcal{J}$ Jacobian matrix. Using this expression and Eq. 5.59 for the mass matrix inverse within Eq. 5.51 leads to the following expression for $\underline{\Lambda}$:

$$\underline{\Lambda} \overset{5.51}{=} \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^* \overset{5.59}{=} \mathcal{B}^*\phi^*H^*(I - H\psi\mathcal{K})^*\mathcal{D}^{-1}(I - H\psi\mathcal{K})H\phi\mathcal{B} \tag{5.62}$$

Using the spatial operator identity [12, 24]

$$(I - H\psi\mathcal{K})H\phi = H\psi \tag{5.63}$$

in Eq. 5.62 leads to the following simpler expression for $\underline{\Lambda}$:

$$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B} \text{ with } \Omega \overset{\triangle}{=} \psi^*H^*\mathcal{D}^{-1}H\psi \in \mathcal{R}^{6n_c \times 6n_c} \tag{5.64}$$

We have arrived at an expression for $\underline{\Lambda}$, that unlike Eq. 5.51, involves neither the mass matrix inverse nor the node's Jacobian matrix! We refer to Ω as the *Extended Operational Space Compliance Matrix*. This terminology is based on Eq. 5.64 which shows that the OSCM, $\underline{\Lambda}$ can be obtained by a reducing transformation of the full, all body Ω matrix by the $\mathcal{B}$ pick-off operator involving just the matrix sub-blocks associated with the parent links of the nodes. From its definition, it is clear that Ω is a symmetric and positive semi-definite since $\mathcal{D}^{-1}$ is a symmetric positive-definite matrix.

While the explicit computation of $\mathcal{M}^{-1}$ or $\mathcal{J}$ is not needed to obtain $\underline{\Lambda}$, the direct evaluation of Eq. 5.64 still remains of $O(\mathcal{N}^3)$ complexity due to the need for carrying out the multiple matrix/matrix products. The next section shows that these matrix/matrix products can be avoided by exploiting a decomposition of the Ω matrix.

Decomposition of $\boldsymbol{\Omega}$

The following lemma describes a decomposition of Ω into simpler component terms and an expression for its block elements. The $\mathcal{E}_{\psi}^*$ and $\psi()$ terms used below are defined in references [12, 23]. Furthermore, $\wp(k)$ denotes the parent link for the kth link, and $i \prec j$ notation implies that the j^{th} link is an ancestor of the ith link in the tree.

Lemma 1 (Decomposition of Ω) Ω *can be decomposed into the following disjoint sum:*

$$\Omega = \Upsilon + \tilde{\psi}^{*}\Upsilon + \Upsilon\tilde{\psi} + R \quad \textit{where} \quad R \triangleq \sum_{\substack{\forall i,j:\, i \not\prec\not\succ j \\ k=\wp(i,j)}} \mathrm{e}_i \psi^{*}(k,i) Y(k) \psi(k,j) \mathrm{e}_j^{*} \tag{5.65}$$

$\Upsilon \in \mathcal{R}^{6n_c \times 6n_c}$ *is a block-diagonal operator, referred to as the operational space compliance kernel, satisfying the following backward Lyapunov equation:*

$$H^{*}\mathcal{D}^{-1}H = \Upsilon - \mathrm{diagOf}\left\{\mathcal{E}_{\psi}^{*}\Upsilon\mathcal{E}_{\psi}\right\} \tag{5.66}$$

$\mathrm{diagOf}\left\{\mathcal{E}_{\psi}^{*}\Upsilon\mathcal{E}_{\psi}\right\}$ *represents just the block-diagonal part of the (generally non block-diagonal)* $\mathcal{E}_{\psi}^{*}\Upsilon\mathcal{E}_{\psi}$ *matrix. The* 6×6 *dimensional, symmetric, positive semi-definite* $\Upsilon(k)$ *diagonal matrices satisfy the following parent/child recursive relationship:*

$$\Upsilon(k) = \psi^{*}(\wp(k),k)\Upsilon(\wp(k))\psi(\wp(k),k) + H^{*}(k)\mathcal{D}^{-1}(k)H(k) \tag{5.67}$$

This relationship forms the basis for the following $O(\mathcal{N})$ *base-to-tips scatter recursion for computing the* $\Upsilon(k)$ *diagonal elements:*

$$\left\{\begin{array}{l} \textbf{for } \textit{all nodes } \boldsymbol{k} \;\; \textit{(base-to-tips scatter)} \\ \quad \Upsilon(k) = \psi^{*}(\wp(k),k)\Upsilon(\wp(k))\psi(\wp(k),k) + H^{*}(k)\mathcal{D}^{-1}(k)H(k) \\ \textbf{end loop} \end{array}\right. \tag{5.68}$$

While Υ *defines the block-diagonal elements of* Ω*, the following recursive expressions describe its off-diagonal terms:*

$$\Omega(i,j) = \left\{\begin{array}{ll} \Upsilon(i) & \textit{for } i = j \\ \Omega(i,k)\psi(k,j) & \textit{for } i \succeq k \succ j, \quad k = \wp(j) \\ \Omega^{*}(j,i) & \textit{for } i \prec j \\ \Omega(i,k)\psi(k,j) & \textit{for } i \not\succ j, \;\; j \not\succ i, \;\; k = \wp(i,j) \end{array}\right. \tag{5.69}$$

Proof See [12, 23].

Equation 5.65 shows that Ω can be decomposed into the sum of simpler terms consisting of the block diagonal Υ, the upper-triangular $\tilde{\psi}^{*}\Upsilon$, the lower triangular $\Upsilon\tilde{\psi}$, and the sparse R matrices. Furthermore, Eq. 5.69 reveals that all of the block-elements of $\Omega(i,j)$ can be obtained from the $\Upsilon(i)$ elements of the Υ block-diagonal operational space compliance kernel.

From the $\underline{\Lambda} = \mathcal{B}^{*}\Omega\mathcal{B}$ expression, and the sparse structure of $\mathcal{B}$, it is clear that only a subset of the elements of Ω are needed to compute $\underline{\Lambda}$. The $\mathcal{B}$ pick-off operator has one column for each of the nodes, with each such column having only a single non-zero 6×6 matrix entry at the kth parent link slot. Only as many elements of Ω as there are elements in $\underline{\Lambda}$ are needed. Thus, just $n_c \times n_c$ number of 6×6 sub-block

matrices of Ω are required. In view of the symmetry of the matrices, we actually need just $n_c(n_c+1)/2$ such sub-block matrices. The overall complexity of this algorithm is linearly proportional to the number of degrees of freedom, and a quadratic function of the number of nodes. This is much lower than the $O(\mathcal{N}^3)$ complexity implied by Eq. 5.51.

References

1. Stewart D, Trinkle JC (2000) An implicit time-stepping scheme for rigidbody dynamics with Coulomb friction. In: Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065) (IEEE, 2000), pp 162–169
2. Trinkle JC (2003) In: ASME international design engineering technical conference, Chicago
3. Anitescu M, Potra FA (1997) Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. Nonlinear Dyn 14(3):231–247
4. Pfeiffer F (2005) Mechanical system dynamics. Springer, Berlin
5. Jain A, Crean C, Kuo C, von Bremen H, Myint S (2012) Minimal coordinateformulation of contact dynamics in operational space. In: Robotics science and systems, Sydney
6. Tasora A, Anitescu M (2011) A matrix-free cone complementarity approach forsolving large-scale, nonsmooth, rigid body dynamics. Comput Methods Appl Mech Eng 200(5–8):439–453. http://linkinghub.elsevier.com/retrieve/pii/S0045782510001970
7. Todorov E (2010) Implicit nonlinear complementarity: A new approach to contactdynamics. In: 2010 IEEE international conference on robotics and automation vol 5. IEEE, Anchorage, Alaska, pp 2322–2329
8. Yamane K, Nakamura Y (2009) A numerically robust LCP solver for simulatingarticulated rigid bodies in contact. In: Robotics science and systems IV. MIT Press, Cambridge, pp 89–104
9. Jain A (2013) Minimal coordinates formulation of contact dynamics. In: Multibody dynamics 2013, ECCOMAS thematic conference, Zagreb, Croatia
10. Walker MW, Orin DE (1982) Efficient dynamic computer simulation of robotic mechanisms. ASME J Dyn Syst Measur Control 104(3):205–211
11. Featherstone R (2008) Rigid body dynamics algorithms. Springer, Berlin
12. Jain A (2011) Robot and multibody dynamics: analysis and algorithms. Springer, Berlin
13. Jain A, Crean C, Kuo C, Quadrelli MB (2012) Efficientconstraint modeling for closed-chain dynamics. In: The 2nd joint international conference on multibody system dynamics, Stuttgart
14. Khatib O (1988) Object manipulation in a multi-effector system. In: 4th international symposium on robotics research. Santa Cruz, pp 137–144
15. Khatib O (1987) A unified approach for motion and force control of robot manipulators: the operational space formulation. IEEE J Robot Autom RA-3(1):43–53
16. Kreutz-Delgado K, Jain A, Rodriguez G (1992) Recursive formulation of operational space control. Int J Robot Res 11(4):320–328
17. Featherstone R (2010) Exploiting sparsity in operational-space dynamics. Int J Robot Res 29(1992):1353–1368
18. Bhalerao KD, Anderson KS, Trinkle JC (2009) A recursive hybrid time-stepping scheme for intermittent contact in multi-rigid-body dynamics. J Comput Nonlinear Dyn 4(4):041010
19. Featherstone R (1999) A divide-and-conquer articulated-body algorithmfor parallel O(log(n)) calculation of rigid-body dynamics. Part 2:basic algorithm. Int J Robot Res 18(9):867–875
20. Bullet Physics Library (2013) http://bulletphysics.org
21. The PATH Solver (2012) http://pages.cs.wisc.edu/ferris/path.html
22. Studer CW (2008) Augmented time-stepping integration of non-smooth dynamical systems. PhD thesis, ETH Zurich

23. Jain A (2011) Graph theoretic foundations of multibody dynamics part II: analysis and algorithms. Multibody Syst Dyn 26(3):335–365
24. Rodriguez G, Jain A, Kreutz-Delgado K (1991) A spatial operator algebra for manipulator modeling and control. Int J Robot Res 10(4):371

Chapter 6
Modelling and Integration Concepts of Multibody Systems on Lie Groups

Andreas Müller and Zdravko Terze

Abstract Lie group integration schemes for multibody systems (MBS) are attractive as they provide a coordinate-free and thus singularity-free approach to MBS modeling. The Lie group setting also allows for developing integration schemes that preserve motion integrals and coadjoint orbits. Most of the recently proposed Lie group integration schemes are based on variants of generalized alpha Newmark schemes. In this chapter constrained MBS are modeled by a system of differential-algebraic equations (DAE) on a configuration space being a subvariety of the Lie group $SE(3)^n$. This is transformed to an index 1 DAE system that is integrated with Munthe-Kaas (MK) integration scheme. The chapter further addresses geometric integration schemes that preserve integrals of motion. In this context, a non-canonical Lie-group Störmer-Verlet integration scheme with direct $SO(3)$ rotational update is presented. The method is 2nd order accurate and it is angular momentum preserving for a free-spinning body. Moreover, although being fully explicit, the method achieves excellent conservation of the angular momentum of a free rotational body and the motion integrals of the Lagrangian top. A higher-order coadjoint-preserving integration scheme on $SO(3)$ is also presented. This method exactly preserves spatial angular momentum of a free body and it is particularly numerically efficient.

A. Müller (✉)
University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai, China
e-mail: andreas.muller@ieee.org

Z. Terze
Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb, Croatia
e-mail: zdravko.terze@fsb.hr

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_6,

6.1 Introduction

Multibody systems (MBS) are commonly modeled by motion equations evolving on the parameter manifold corresponding to a specific parameterization in terms of angles and displacements. Since the motion of rigid bodies, and thus of a MBS, evolves on a Lie group the kinematics and dynamics of a MBS can be modeled coordinate-free as a dynamic system on a Lie group, which avoids the well-known problem of parameterization singularities. This allows for application of Lie group integration methods that are inherently coordinate-free. The original application of Lie group integration schemes to rigid body dynamics was the dynamics of a rotating body. To this end the Munthe-Kaas (MK) method based on a Runge-Kutta scheme is used [9, 14, 22, 25–27, 29], and later an amended Newmark-Verlet scheme was applied [15]. In this case the rotation group $SO(3)$ is readily identified as configuration space (c-space). From a MBS perspective this corresponds to a 'minimal coordinate' or 'relative coordinate' formulation in the sense that the body is free to move on $SO(3)$. From a more general point of view, considering the rigid body as being constrained by a spherical joint to rotate about a fixed point, $SO(3)$ is the isotropy group of the joint. Hence this is the Lie group of relative motions of two bodies connected by a spherical joint. This 'relative coordinate' approach applies to general tree-topology MBS where all joints describe motion subgroups of $SE(3)$. Along this line the MK scheme was applied to the dynamics of unconstrained serial manipulators in [30]. Instead of resorting to the motion (isotropy) groups of joints, the motion of individual bodies can be modeled on $SE(3)$ and a general MBS with topological loops be modeled by imposing corresponding constraints. This is equivalent to the 'absolute coordinate' formulation. That is, the configuration of an MBS comprising n rigid bodies is a subvariety of $SE(3)^n$. Following the common MBS modeling approach, rigid body configurations are frequently assumed to belong to the Lie group $SO(3) \times \mathbb{R}^3$, however. Therewith in [33, 34] the MBS motion equations are formulated as an index 1 differential-algebraic equations (DAE) and solved with a MK scheme. In [4, 5, 19] a generalized alpha/Newmark scheme is used to integrate the motion equations of constrained MBS formulated on $SO(3) \times \mathbb{R}^3$. Since Lie group integrators exploit the geometry of the c-space, replacing $SE(3)$ by $SO(3) \times \mathbb{R}^3$ has consequences for the performance. This issue is addressed in [23], and it is shown that in specific cases, $SE(3)$ achieves perfect constraint satisfaction whereas in the general case both c-spaces lead to equivalent results. In the first part of this chapter a DAE index 1 MBS model on $SE(3)$ is introduced and its integration with a MK scheme described.

Besides a coordinate-free description the modeling of MBS on a Lie group provides a setting for structure preserving integration schemes. A non-canonical Lie-group Störmer-Verlet integration scheme on $SO(3)$ is presented in the second part of this chapter. The method is 2nd order accurate, it is angular momentum preserving, and it does not introduce an energy drift. It is shown that, although being fully explicit, the method conserves the angular momentum of a freely rotating body and the motion integrals of the Lagrangian top better than established conserving algo-

rithms. Finally higher-order integration scheme is presented that preserves coadjoint orbits. The method exactly preserves spatial angular momentum of a free body and is numerically efficient since the rotation and momentum update use the same incremental rotation vector.

6.2 Lie Group Integration Schemes for Multibody Systems

6.2.1 Lie Group Modeling of Multibody Systems

The motion equations of a holonomically constrained multibody system (MBS) comprising n rigid bodies in terms of non-holonomic velocities attain the form

$$\mathbf{M}(q)\dot{\mathbf{V}} + \mathbf{J}^T\boldsymbol{\lambda} = \mathbf{Q}\,(q, \mathbf{V}, t) \tag{6.1a}$$

$$\dot{q} = q \cdot \mathbf{V} \tag{6.1b}$$

$$g(q) = \mathbf{0}. \tag{6.1c}$$

Adopting the 'absolute coordinate' approach the configuration $q = (\mathbf{C}_1, \ldots, \mathbf{C}_n) \in G$ of a MBS is given in terms of the 'absolute' configuration $\mathbf{C}_i \in SE(3)$ of the n bodies, where $G = SE(3)^n$ is the 6-dimensional Lie group representing the rigid body configuration. The MBS velocity $\mathbf{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_n) \in \mathfrak{g}$ consists of the body-fixed velocities $\mathbf{V}_i = (\boldsymbol{\omega}_i, \mathbf{v}_i)\,, i = 1, \ldots, n$, where $\mathfrak{g} = se\,(3)^n$ is the Lie algebra of G. The Eqs. (6.1a–6.1c) form a system of differential-algebraic equations (DAE) on $G \times \mathfrak{g}$ with geometric constraints (6.1c). In the dynamic motion equations (6.1a) $\boldsymbol{\lambda}$ are Lagrange multipliers and $\mathbf{J}$ is the constraint Jacobian. The system (6.1b), relating the MBS velocity and motion, is called the *kinematic reconstruction equations*. Its solution is the motion $q(t)$ of the MBS. Traditionally, describing the MBS configuration in terms of position vectors $\mathbf{r}_i$ and certain rotation parameters $\boldsymbol{\theta}_i$, the kinematic reconstruction equations assume the decoupled form $\boldsymbol{\omega}_i = \mathbf{B}_i\,(\mathbf{q})\,\dot{\boldsymbol{\theta}}_i$ and $\mathbf{v}_i = \dot{\mathbf{r}}_i, i = 1, \ldots, n$. In the Lie group approach using $SE(3)$ the geometry of rigid body motions is respected. This is not the case when rotations and translations are treated independently. It has been shown [23] that this leads to constraint violations.

Background on the Lie group modeling of rigid body motions can be found in the books by Selig [31] and Murray et al. [28] for instance.

6.2.2 Configuration Space Lie Group

Body-fixed reference frames attached to the n rigid bodies represent the configuration of the MBS. The configuration of body i with respect to a world-fixed inertial frame (IFR) is described by the pair $C_i = (\mathbf{R}_i, \mathbf{r}_i)$, which can be represented as matrix

$$\mathbf{C}_i = \begin{pmatrix} \mathbf{R}_i & \mathbf{r}_i \\ \mathbf{0} & 1 \end{pmatrix} \tag{6.2}$$

where $\mathbf{r}_i \in \mathbb{R}^3$ is the position vector of its origin and $\mathbf{R}_i \in SO(3)$ is the absolute rotation matrix. Therewith the combination of two successive rigid-body motions is given by

$$\mathbf{C}_2\mathbf{C}_1 = \begin{pmatrix} \mathbf{R}_2\mathbf{R}_1 & \mathbf{r}_2 + \mathbf{R}_2\mathbf{r}_1 \\ \mathbf{0} & 1 \end{pmatrix}. \tag{6.3}$$

Such rigid body motions constitute the 6-dimensional Lie group $SE(3) = SO(3) \ltimes \mathbb{R}^3$ of isometric orientation preserving transformations of E^3.

The *ambient configuration space* of an MBS consisting of n rigid bodies is the $6n$-dimensional Lie group

$$G := SE(3)^n \tag{6.4}$$

where $q = (\mathbf{C}_1, \ldots, \mathbf{C}_n) \in G$ represents the configuration of n bodies in a coordinate-free way. Multiplication is componentwise $q' \cdot q'' = (\mathbf{C}_1'\mathbf{C}_1'', \ldots, \mathbf{C}_n'\mathbf{C}_n'')$, inherited from $SE(3)$. The inverse is $q^{-1} = (\mathbf{C}_1^{-1}, \ldots, \mathbf{C}_n^{-1})$.

The Lie group $SE(3)$ is generated from its Lie algebra $se(3)$ via the exp mapping. Since any rigid body motion is a screw motion the exp mapping gives a finite screw motion corresponding to an instantaneous screw motion $\mathbf{X}(t)$. Explicitly this is

$$\mathbf{X} = (\boldsymbol{\xi}, \boldsymbol{\eta}) \longmapsto \exp\widehat{\mathbf{X}} = \begin{pmatrix} \exp\widehat{\boldsymbol{\xi}} & \frac{1}{\|\boldsymbol{\xi}\|^2}\left(I - \exp\widehat{\boldsymbol{\xi}}\right)(\boldsymbol{\xi} \times \boldsymbol{\eta}) + h\boldsymbol{\xi} \\ 0 & 1 \end{pmatrix} \tag{6.5}$$

where

$$\exp\widehat{\boldsymbol{\xi}} = \mathbf{I} + \frac{\sin\|\boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|}\widehat{\boldsymbol{\xi}} + \frac{1 - \cos\|\boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|}\widehat{\boldsymbol{\xi}}^2 \tag{6.6}$$

is the exponential mapping on $SO(3)$. $\mathbf{X} = (\boldsymbol{\xi}, \boldsymbol{\eta}) \in \mathbb{R}^6$ is the instantaneous screw coordinate vector describing the motion of a rigid body. The six components serve as independent local canonical coordinates on $SE(3)$. It is important to observer that they are not just the scaled rotation axis and translation vector. $\boldsymbol{\xi}$ is in fact the Euler-Rodrigues vector but $\boldsymbol{\eta}$ is not the displacement vector.

Let the bodies of the MBS be subjected to a system of m scleronomic geometric constraints

$$g(q) = \mathbf{0} \tag{6.7}$$

defined by the constraint mapping $g: G \rightarrow \mathbb{R}^m$. Then the *configuration space (c-space)* of the MBS is the variety

$$\mathcal{V} := \{q \in G | g(q) = \mathbf{0}\}. \tag{6.8}$$

6.2.3 State Space Lie Group

A rigid body motion is a curve $\mathbf{C}(t)$ in $SE(3)$ The *body-fixed velocity* of body i, expressed by the vector $\mathbf{V}_i = (\boldsymbol{\omega}_i, \mathbf{v}_i)^T \in \mathbb{R}^6$ where $\boldsymbol{\omega}_i$ is the body-fixed angular velocity and $\mathbf{v}_i$ is the translation velocity vector expressed in the body-fixed RFR, is defined as

$$\widehat{\mathbf{V}}_i := \mathbf{C}_i^{-1}\dot{\mathbf{C}}_i = \begin{pmatrix} \widehat{\boldsymbol{\omega}}_i & \mathbf{v}_i \\ \mathbf{0} & 0 \end{pmatrix} \in se(3) \tag{6.9}$$

with $se(3)$ being the Lie algebra of $SE(3)$. $\widehat{\boldsymbol{\omega}}_i := \mathbf{R}_i^T\dot{\mathbf{R}}_i \in so(3)$ is the skew symmetric (cross product) matrix associated to the vector $\boldsymbol{\omega}$. Hence to any twist coordinate vector is assigned a $se(3)$-matrix via the 'hat' operator.

The right-translated differential of the exp mapping, $\mathrm{dexp} : se(3)\times se(3) \to se(3)$, can be introduced as $\mathrm{dexp}_{\widehat{\mathbf{X}}}\dot{\widehat{\mathbf{X}}} = \dot{\mathbf{C}}\mathbf{C}^{-1}$, with $\mathbf{C} = \exp\widehat{\mathbf{X}}$. Replacing $\mathbf{X}$ with $-\mathbf{X}$ leads to the expression for the body-fixed twist:

$$\widehat{\mathbf{V}} = \mathrm{dexp}_{-\widehat{\mathbf{X}}}\dot{\widehat{\mathbf{X}}}. \tag{6.10}$$

Hence the vector of body-fixed velocity is given in terms of the time derivative of the screw coordinates $\mathbf{X}$. These are the kinematic reconstruction equations (6.1b) expressed in terms of screw coordinates, that must be solved in order to recover the finite motion from the velocity field. They will be the basis for the Munthe-Kaas method in Sect. 6.2.5.

The inverse of this mapping, which is subsequently needed, is in vector representation given by [31]

$$\begin{aligned}\mathbf{dexp}_{\widehat{\mathbf{X}}}^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{ad}_{\mathbf{X}} + \left(\frac{2}{\|\boldsymbol{\xi}\|^2} + \frac{\|\boldsymbol{\xi}\| + 3\sin\|\boldsymbol{\xi}\|}{4\|\boldsymbol{\xi}\|(\cos\|\boldsymbol{\xi}\| - 1)}\right)\mathbf{ad}_{\mathbf{X}}^2 \\ + \left(\frac{1}{\|\boldsymbol{\xi}\|^4} + \frac{\|\boldsymbol{\xi}\| + \sin\|\boldsymbol{\xi}\|}{4\|\boldsymbol{\xi}\|^3(\cos\|\boldsymbol{\xi}\| - 1)}\right)\mathbf{ad}_{\mathbf{X}}^4\end{aligned} \tag{6.11}$$

with $\mathbf{X} = (\boldsymbol{\xi}, \boldsymbol{\eta})$. An alternative form can be determined, as reported e.g. in [30] , is

$$\mathbf{dexp}_{\mathbf{X}}^{-1} = \begin{pmatrix} \mathbf{dexp}_{\boldsymbol{\xi}}^{-1} & \mathbf{0} \\ \mathbf{U} & \mathbf{dexp}_{\boldsymbol{\xi}}^{-1} \end{pmatrix} \tag{6.12}$$

with

$$\mathbf{U}(\mathbf{X}) = \frac{1-\gamma}{\|\boldsymbol{\xi}\|^2}\left(\widehat{\boldsymbol{\eta}}\widehat{\boldsymbol{\xi}} + \widehat{\boldsymbol{\xi}}\widehat{\boldsymbol{\eta}}\right) + \frac{h_{\mathbf{X}}}{\|\boldsymbol{\xi}\|^3}\left(\frac{1}{\beta} + \gamma - 2\right)\widehat{\boldsymbol{\xi}}^2 - \frac{1}{2}\widehat{\boldsymbol{\eta}} \tag{6.13}$$

$\beta := \frac{4}{\|\boldsymbol{\xi}\|^2}\sin^2\frac{\|\boldsymbol{\xi}\|}{2}$, $\gamma := \frac{2}{\|\boldsymbol{\xi}\|}\cot\frac{\|\boldsymbol{\xi}\|}{2}$, and the pitch $h = \boldsymbol{\xi}\cdot\boldsymbol{\eta}/\|\boldsymbol{\xi}\|^2$. Notice that for pure rotation, i.e. $h_{\mathbf{X}} = 0$, (6.13) simplifies. In (6.12) $\mathbf{dexp}_{\boldsymbol{\xi}}^{-1}$ is the inverse of the differential of the exp mapping (6.6) on $SO(3)$ for $\mathbf{R} = \exp\widehat{\boldsymbol{\xi}}$ [6]

$$\mathbf{dexp}_{\xi}^{-1} = \mathbf{I} - \frac{1}{2}\widehat{\boldsymbol{\xi}} + \left(1 - \frac{\|\boldsymbol{\xi}\|}{2}\cot\frac{\|\boldsymbol{\xi}\|}{2}\right)\frac{\widehat{\boldsymbol{\xi}}^2}{\|\boldsymbol{\xi}\|^2}. \tag{6.14}$$

Notice finally, that therewith the exp mapping (6.5) can be expressed as

$$\mathbf{X} = (\boldsymbol{\xi}, \boldsymbol{\eta}) \longmapsto \exp\widehat{\mathbf{X}} = \begin{pmatrix} \exp\widehat{\boldsymbol{\xi}} & \mathbf{dexp}_{\xi}\boldsymbol{\eta} \\ \mathbf{0} & 1 \end{pmatrix}. \tag{6.15}$$

The velocities of the bodies of the MBS are collectively represented by $\mathbf{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_n) \in \mathbb{R}^{6n}$. Making use of the isomorphism of $se^n(3)$ and $\mathbb{R}^{6n}$ the body-fixed velocities are determined as $\widehat{\mathbf{V}} = q^{-1}\dot{q}$, denoting $\widehat{\mathbf{V}} = (\widehat{\mathbf{V}}_1, \ldots, \widehat{\mathbf{V}}_n)$. This allows to introduce the *ambient state space* of the MBS as

$$S := SE(3)^n \times \mathbb{R}^{6n}, \tag{6.16}$$

which is a $6 \cdot 2 \cdot n$-dimensional Lie group. This is the left-trivialized tangent bundle of G. The state of the MBS is then represented by $X = (q, \mathbf{V}) = (\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{V}_1, \ldots, \mathbf{V}_n) \in S$. The multiplication is understood componentwise: $X' \cdot X'' = \left(\mathbf{C}'_1\mathbf{C}''_1, \ldots, \mathbf{C}'_n\mathbf{C}''_n, \mathbf{V}'_1 + \mathbf{V}''_1, \ldots, \mathbf{V}'_n + \mathbf{V}''_n\right)$.

The Lie algebra of the Lie group S is

$$\mathfrak{s} := se(3)^n \times \mathbb{R}^{6n}, \tag{6.17}$$

with elements $x = (\mathbf{V}_1, \ldots, \mathbf{V}_n, \mathbf{A}_1, \ldots, \mathbf{A}_n) \in \mathfrak{s}$, which is isomorphic to the tangent space of S via left translation, i.e. $\dot{X} = X \cdot x$. Addition in $\mathfrak{s}$ is also componentwise: $x' + x'' = (\mathbf{V}'_1 + \mathbf{V}''_1, \ldots, \mathbf{V}'_n + \mathbf{V}''_n, \mathbf{A}'_1 + \mathbf{A}''_1, \ldots, \mathbf{A}'_n + \mathbf{A}''_n)$. The exponential mapping on the ambient state space Lie group is

$$\exp x = (\exp\mathbf{V}_1, \ldots, \exp\mathbf{V}_n, \mathbf{A}_1, \ldots, \mathbf{A}_n) \in S \tag{6.18}$$

with (6.5). Its right-translated differential $\mathrm{dexp} : \mathfrak{s} \times \mathfrak{s} \to \mathfrak{s}$ is

$$\mathrm{dexp}_{x'}x'' = (\mathrm{dexp}_{\mathbf{V}'_1}\mathbf{V}''_1, \ldots, \mathrm{dexp}_{\mathbf{V}'_n}\mathbf{V}''_n, \mathbf{A}''_1, \ldots, \mathbf{A}''_n) \tag{6.19}$$

where for the first n components $\mathrm{dexp} : se(3) \times se(3) \to se(3)$ is the right-translated differential on $SE(3)$.

Time differentiation of the geometric constraints (6.7) yields the corresponding velocity constraints

$$\mathbf{J}(q)\,\mathbf{V} = \mathbf{0} \tag{6.20}$$

where $\mathbf{J}(q)\colon \mathbb{R}^{6n} \to \mathbb{R}^m$ is the Jacobian of g in vector representation. Together with the geometric constraints they define the *MBS state space*

$$\mathcal{S} := \{X = (q, \mathbf{V}) \in S | g(q) = 0, \mathbf{J}(q)\,\mathbf{V} = 0\}. \tag{6.21}$$

The state space S being a Lie group admits to apply Lie group integration schemes.

6.2.4 Motion Equations of Constrained MBS in Lie Group Descriptor Form

The motion equations (6.1a–6.1c) form an index 3 system of differential-algebraic equations (DAE) on the state space Lie group $\mathcal{S}$. The latter factors into the vector space $\mathbb{R}^{6n}$ and the the Lie group G. In particular the kinematic reconstruction equations (6.1b) form a system on G consisting of the n equations $\dot{\mathbf{C}}_i = \mathbf{C}_i \widehat{\mathbf{V}}_i$. The dynamic equations (6.1a) evolve on the vector space $\mathbb{R}^n$.

A common approach in MBS modeling is to transform the system (6.1a) together with (6.1c) to the index 1 system

$$\begin{pmatrix} \mathbf{M} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{V}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{Q} \\ \boldsymbol{\eta} \end{pmatrix} \tag{6.22}$$

using the acceleration constraints $\mathbf{J}(q) \cdot \dot{\mathbf{V}} = \boldsymbol{\eta}(q, \mathbf{V})$. For a given state $X = (q, \mathbf{V}) \in \mathcal{S}$ the system (6.22) and thus (6.1a) can be solved for $\dot{\mathbf{V}}$. If $\dot{\mathbf{V}}$ is a solution of (6.22), then (6.1a, 6.1c) is equivalent to the ODE system

$$\dot{X} = XF(t, X) \tag{6.23}$$

on the state space S, where the mapping $F: \mathbb{R} \times S \to \mathfrak{s}$ is introduced as $F(t, X) = (\mathbf{V}, \dot{\mathbf{V}})$ with a solution $\dot{\mathbf{V}}$ of (6.22). The equivalence follows from $XF(t, X) = (q \cdot \mathbf{V}, \dot{\mathbf{V}})$. Evaluation of $XF(t, X)$ thus amounts to solving (6.22) for $\dot{\mathbf{V}}$ and evaluating (6.1b).

6.2.5 Munthe-Kaas Method for Constrained MBS Dynamics

As the dynamical system evolves on a Lie group its solution can be expressed in the form $X(t) = X_0 \exp \Phi(t)$. Hence at the integration step i the original system (6.23) can be replaced by the system

$$\dot{\Phi}^{(i)} = \operatorname{dexp}^{-1}_{-\Phi^{(i)}} F(t, X_{i-1} \exp \Phi^{(i)}), \ t \in [t_{i-1}, t_i], \ \text{with } \Phi^{(i)}(t_{i-1}) = 0 \tag{6.24}$$

with initial condition X_{i-1}, and be solved with a numerical integration scheme. The Munthe-Kaas (MK) method [12] uses a Runge-Kutta (RK) scheme to determine a solution $\Phi^{(i)}(t_i)$ that leads to a numerical solution $X_i := X_{i-1} \exp \Phi^{(i)}(t_i)$ of (6.23). The $\Phi^{(i)}$ constitute local coordinates on the state space that are valid in a neighborhood of X_{i-1}. An s-stage MK scheme at time step i follows immediately from the RK method as

$$X_i := X_{i-1} \exp \Phi^{(i)}, \quad \Phi^{(i)} := h \sum_{j=1}^{s} b_j k_j \tag{6.25}$$

$$k_j := \operatorname{dexp}^{-1}_{-\Psi_j} F\left(t_{i-1} + c_j h, X_{i-1} \exp \Psi_j\right), \quad \Psi_j := h \sum_{l=1}^{j-1} a_{jl} k_l, \ \Psi_1 = 0,$$

where a_{jl}, b_j, and c_j are the Butcher coefficients of the s -stage RK method, and k_j, $\Psi_j \in \mathfrak{s}$. MK methods can be applied to ODE on Lie groups. Using the formulation (6.23) the MK can also be applied to the time integration of dynamics equations of constrained MBS.

Remark 1 While it is clear that $SE(3)$ is the c-space Lie group of a rigid body, all original MBS formulations implicitly use the Lie group $SO(3) \times \mathbb{R}^3$ since it is customary to use rotation angles and displacement vectors as parameters which are clearly not screw parameters. This leads to an artificial decoupling of rotational and translational motion. This is reflected by the dexp mapping which for $SO(3) \times \mathbb{R}^3$ *is*

$$\operatorname{dexp}_{\mathbf{X}}(\mathbf{Y}) = (\operatorname{dexp}_{\boldsymbol{\xi}} \boldsymbol{\eta}, \mathbf{v}) \tag{6.26}$$

for $\mathbf{X} = (\boldsymbol{\xi}, \mathbf{u})$ and $\mathbf{Y} = (\boldsymbol{\eta}, \mathbf{v})$. Even more, also the recently proposed Lie group generalized alpha schemes [4, 5] and Lie group DAE formulation [33] use $SO(3) \times \mathbb{R}^3$ as c-space. As consequence the configuration update step in (6.25) does not respect the screw motion that is encoded in the velocity $\mathbf{V}$, which leads in particular to constraint violations as shown in [23]. However, it is shown in [23] that the difference of the two c-spaces becomes significant for the satisfaction of joint constraints only when a rigid body is connected to the ground by lower-pair joints. In this case the joint constraints are perfectly satisfied independently of the integration accuracy and step size.

Remark 2 It is well-known that the numerical solution of the index 1 formulation (6.23) may not stay in $\mathcal{V}$. Nevertheless, the index 1 formulation (6.22) is convenient for treating the dynamics of constrained MBS. For the classical vector space formulation several constraint stabilization methods have been proposed to cope with the drift phenomenon (an overview can be found in [2]). Within the Lie group formulation these have to be amended as reported in [24, 34]. In [24] a constraint stabilization method is presented that uses the local coordinates $\Phi^{(i)}$ of the MK scheme. In [34] stabilisation algorithm based on constrained least square minimization algorithm in Lie groups state space is introduced.

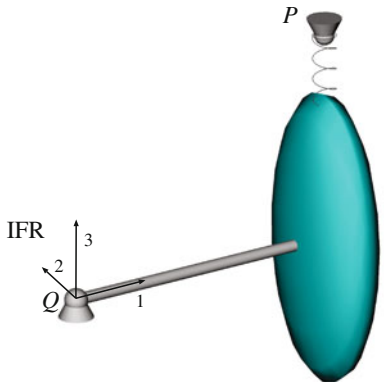


Fig. 6.1 Rigid body constrained to rotate about fixed point subject to an applied spring force

6.2.6 Examples

6.2.6.1 Heavy Top with External Force

The model of a heavy top (Fig. 6.1) consists of a rigid body pivoted to the ground at point Q. A spring is attached between its COM and a space-fixed point P. The spring force is $\mathbf{F}^s = c(\mathbf{p}_0 - \mathbf{r}) + m\mathbf{g}$, with 10 N/mm, and gravity vector $\mathbf{g} = (0, 0, -9.81)^T$, with $\mathbf{p}_0 = (1, 0, 0.5)^T$ m being the space-fixed position vector of the spring suspension point P. The body-fixed force vector (6.30) is $\mathbf{F} = \mathbf{R}^T\mathbf{F}^s$. The inertia of the body's COM is $\boldsymbol{\Theta}_0 = \text{diag}\,(0.36, 0.306, 0.09)$ kg m^2, and its mass $m = 21.6$ kg. Denote with $\mathbf{r}_0 = (-0.5, 0, 0)^T$ m the position vector of the pivot point measured in the body-fixed reference frame. The configuration of the reference frame is represented by $C = (\mathbf{R}, \mathbf{r})$, with rotation matrix $\mathbf{R}$ and $\mathbf{r}$ denoting the position of the COM expressed in the spatial inertial frame (IFR).

The geometric constraints imposed by the spherical joint (pivot) are

$$g(C) = \mathbf{r} - \mathbf{R}\mathbf{r}_0 = \mathbf{0}. \tag{6.27}$$

Time differentiation, and assuming (6.27), yields the velocity constraints (6.20)

$$\left(\widehat{\mathbf{r}}_0 \; -\mathbf{I}\right)\begin{pmatrix}\boldsymbol{\omega}\\ \mathbf{v}\end{pmatrix} = \mathbf{J}\mathbf{V} = \mathbf{0} \tag{6.28}$$

and the acceleration constraints

$$\left(\widehat{\mathbf{r}}_0 \; -\mathbf{I}\right)\begin{pmatrix}\dot{\boldsymbol{\omega}}\\ \dot{\mathbf{v}}\end{pmatrix} = \widehat{\boldsymbol{\omega}}\widehat{\boldsymbol{\omega}}\mathbf{r}_0 + \widehat{\boldsymbol{\omega}}\mathbf{v} \tag{6.29}$$

where $\mathbf{V} = (\boldsymbol{\omega}, \mathbf{v})$ is the body-fixed velocity. The body-fixed Newton-Euler equations w.r.t. to the COM combined with (6.29) yield the overall index 1 DAE system

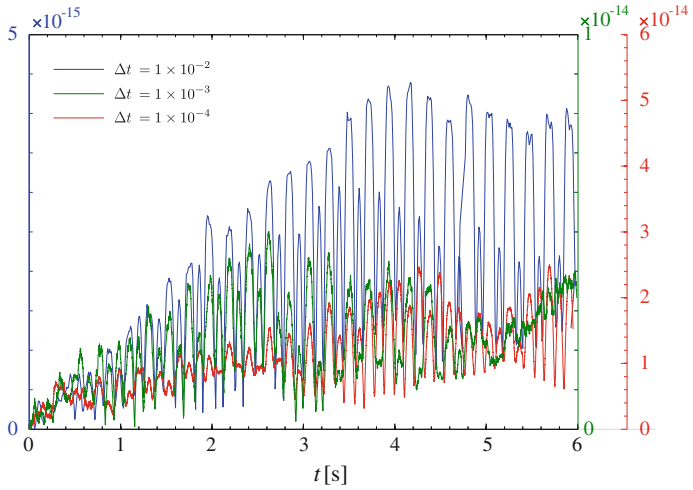


Fig. 6.2 Error ε in the position constraints of the *top*

$$\begin{pmatrix} \boldsymbol{\Theta}_0 & \mathbf{0} & \widehat{\mathbf{r}}_0^T \\ \mathbf{0} & m\mathbf{I} & -\mathbf{I} \\ \widehat{\mathbf{r}}_0 & -\mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{\omega}}\boldsymbol{\Theta}_0\boldsymbol{\omega} \\ \mathbf{F} - m\widehat{\boldsymbol{\omega}}\mathbf{v} \\ \widehat{\boldsymbol{\omega}}\widehat{\boldsymbol{\omega}}\mathbf{r}_0 + \widehat{\boldsymbol{\omega}}\mathbf{v} \end{pmatrix}. \tag{6.30}$$

The motion equations (6.30) were integrated with an MK scheme based on RK4 method for 6 s starting from initial configuration $C_0 = (\mathbf{I}, \mathbf{r}_0)$ and initial angular velocity $\boldsymbol{\omega}_0 = (0, 0, 0.5)^T$ rad/s. The integration method shows 4th order convergence as expected. It is further interesting to observe the exact satisfaction of the position constraints independently of the step size. Figure 6.2 shows the error of the position constraints $\varepsilon(q) := \|g(q)\|$ for step sizes $\Delta t = 10^{-2}, 10^{-3}, 10^{-4}$ s. The satisfaction of the constraints up to computation precision is due to the use of the correct c-space Lie group, namely $SE(3)$, that accounts for rigid body motions. This is discussed in [23].

6.2.6.2 Planar 4-Bar Mechanism

A closed loop planar 4-bar mechanism is considered (Fig. 6.3) comprising three revolute joints and one spherical joint. The geometry is chosen as indicated in the Fig. 6.3 with $L_0 = 0.5$ m.

The initial configuration is shown in Fig. 6.3. Initially the input crank (body 1) rotates with angular velocity $\omega_0 = 10\,\pi$ rad/s. The motion equations are integrated numerically with the MK/RK4 method with step sizes $\Delta t = 10^{-2}, 10^{-3}, 10^{-4}$ s. Figures 6.4 and 6.5 show the satisfaction of joint position constraints. Apparently they are satisfied with computation accuracy for the two joints 1 and 4 that are connected

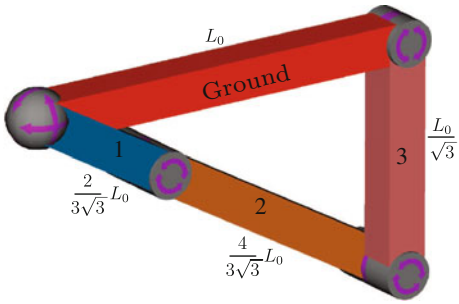


Fig. 6.3 Planar 4-bar mechanism: comprising two revolute joints with parallel axes and a spherical joint

to the ground. The orientation constraints of the revolute joints are exactly satisfied. As mentioned above this phenomenon is due to the use of the correct c-space Lie group as discussed in [23].

6.3 Geometric Schemes that Preserve Integrals of Motion

6.3.1 Coadjoint Modeling of Rotational Dynamics

In the geometric schemes presented in the previous chapters, the kinematic reconstruction equations have been solved on the c-space Lie group while the dynamical equations were discretised via 'classical' vector-space-based numerical methods. Although pursued by several authors [4, 5, 23], this practice does not utilize the geometrical properties of dynamics on the c-space that gives rise to integration algorithms with additional useful properties, such as the preservation of the integrals of motion. Therefore, in the sequel we describe the geometric schemes that extend the coadjoint orbit preserving integration method for $SO(3)$ [8, 16, 18]. Another possibility of constructing the structure-preserving algorithms is to follow the variational approach, see, for example [20, 36], and references cited there.

We start from the Euler equation of free rigid body rotation given as Lie-Poisson's system [21] in the form

$$\dot{\mathbf{y}} = -\hat{\boldsymbol{\omega}}\mathbf{y}, \tag{6.31}$$

where $\mathbf{y} \in \mathcal{R}^3$ represents the angular momentum in the body attached frame and $\hat{\boldsymbol{\omega}} \in so(3)$ is the body angular velocity with $so(3)$ being the Lie algebra of $SO(3)$. By following [13], $\mathbf{y} \in \mathcal{R}^3$can be identified with $\check{\mathbf{y}} \in so^*(3)$, where $so^*(3)$ is the dual space of the Lie algebra $so(3)$. In a more formal form (6.31) can be expressed as the coadjoint operator [14, 18] on the dual space of the Lie-algebra $so^*(3)$ as

$$\dot{\mathbf{y}} = \mathrm{ad}^*_{\hat{\boldsymbol{\omega}}}\mathbf{y} = \hat{\mathbf{y}}\boldsymbol{\omega}(\mathbf{y}), \tag{6.32}$$

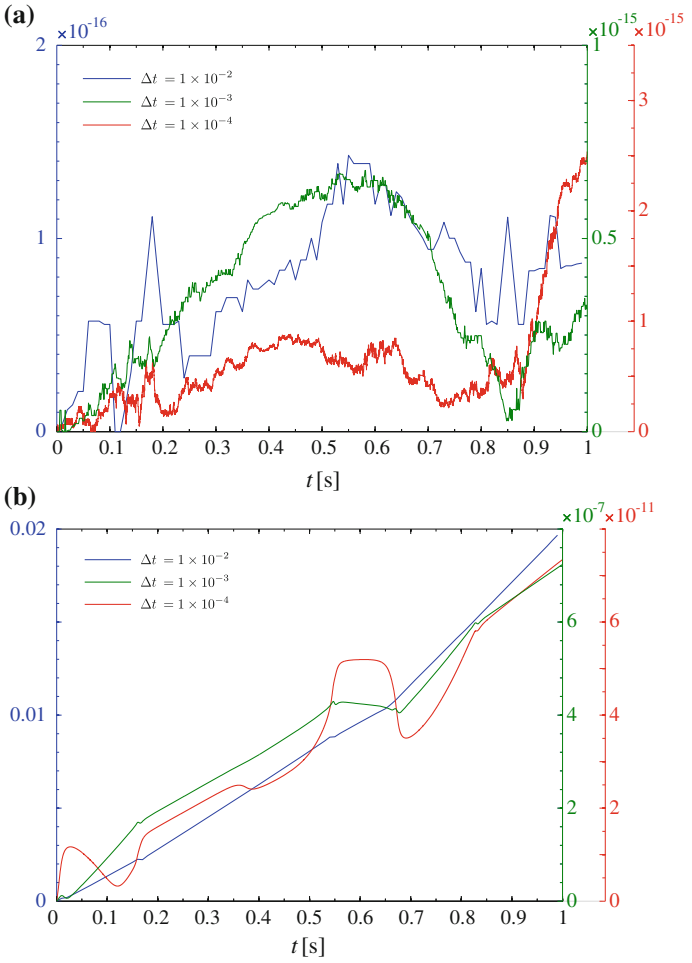


Fig. 6.4 Violation of position constraints of revolute joint 1 (**a**) and 2 (**b**)

where 'ad*' is the dual of the 'ad' operator $\mathrm{ad}_{\hat{\mathbf{a}}}\mathbf{b} = \hat{\mathbf{a}}\mathbf{b}$, which is the commutator

$$\mathrm{ad}_{\hat{\mathbf{a}}}(\hat{\mathbf{b}}) = \hat{\mathbf{a}}\hat{\mathbf{b}} - \hat{\mathbf{b}}\hat{\mathbf{a}} = [\hat{\mathbf{a}}, \hat{\mathbf{b}}], \quad \text{for all } \hat{\mathbf{a}}, \hat{\mathbf{b}} \in so(3), \tag{6.33}$$

in the Lie algebra $so(3)$ identified here with $\mathcal{R}^3$. The solution of the angular momentum equation in the Lie-Poisson form (6.31) can be expressed within each integration step as an action of $SO(3)$ on $\mathcal{R}^3$in the form

$$\begin{aligned} &\mathbf{y}^{n+1} = \mathbf{Q}^{\mathrm{T}}(t)\mathbf{y}^n, \quad n = 0, 1, 2, \ldots, \\ &\mathbf{Q} \in SO(3), \quad t \in \left[t_n, t_{n+1}\right], \end{aligned} \tag{6.34}$$

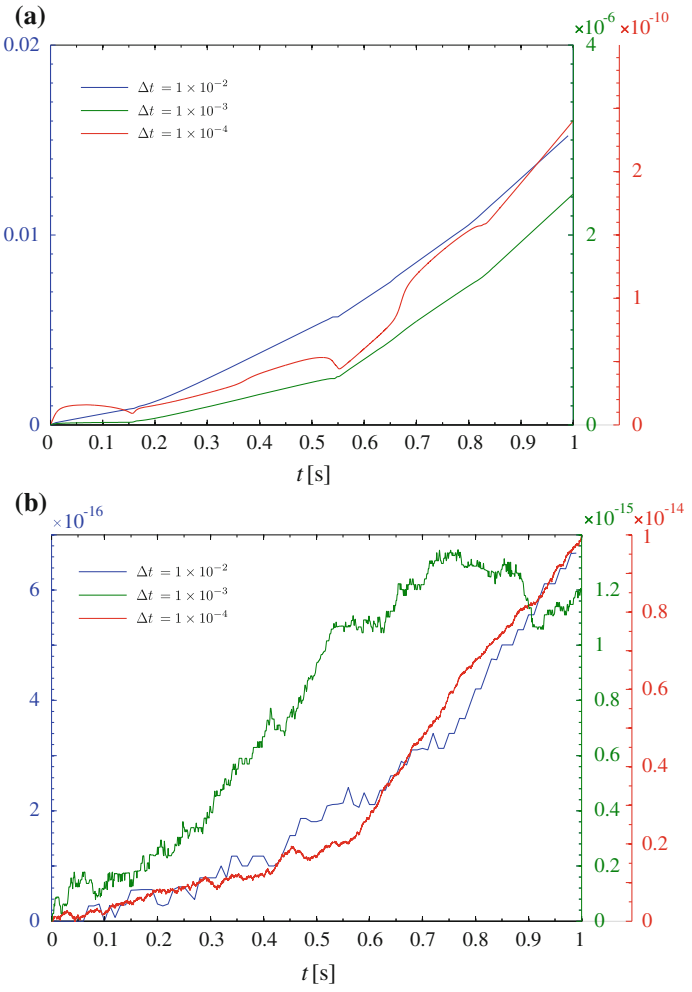


Fig. 6.5 Violation of position constraints of revolute joint 3 (**a**) and spherical joint 4 (**b**)

which leads to solving of ODE on the Lie-group that reads

$$\begin{aligned} \dot{\mathbf{Q}}(t) &= \mathbf{Q}(t)\hat{\boldsymbol{\omega}}\left(\mathbf{y}\left(t\right)\right), \quad t \geq t_n, \\ \mathbf{Q}(t_n) &= I. \end{aligned} \tag{6.35}$$

More formally, the update step (6.34) can be written as coadjoint action [13], denoted 'Ad*', of $SO(3)$ on $\mathcal{R}^3$ in the form [8, 18]

$$\mathbf{y}^{n+1} = \mathrm{Ad}^*_{\mathbf{Q}(t)}\mathbf{y}^n, \tag{6.36}$$

where the relation $\mathrm{Ad}^*_{\mathbf{Q}}\mathbf{y} = \mathbf{Q}^{\mathrm{T}}\mathbf{y}$ is valid and the coadjoint orbit is given as [13]

$$O = \{\mathrm{Ad}^*_{\mathbf{Q}}\mathbf{y} \quad | \quad \mathbf{Q} \in SO(3)\} \subset \mathcal{R}^3. \tag{6.37}$$

The motivation of constructing the update of $\mathbf{y}^n \in \mathcal{R}^3$ in the form of (6.34) [or (6.36)] is the exact preservation of the magnitude of free-body angular momentum in the body attached frame during the step. Indeed, since for the step initial condition $\mathbf{y}^n$ the coadjoint orbit $O_{\mathbf{Y}^n}$ is a sphere of radius $\|\mathbf{y}^n\|$, the magnitude of $\mathbf{y}^n$ will be exactly preserved, independently of the accuracy of the integration method for determining $\mathbf{Q}$ in (6.35).

However, although coadjoint orbits will be preserved independently of the accuracy of determining $\mathbf{Q}$, the rotation matrix $\mathbf{Q}$ has to be determined in order to complete the integration step, i.e. the ODE (6.35) has to be solved. To this end, we will adopt the Muthe-Kaas approach [12, 14, 25] and we seek a solution of (6.35) in the form

$$\mathbf{Q}(t) = \exp(\hat{\boldsymbol{\psi}}(t)), \quad t \geq t_n, \tag{6.38}$$

where the closed form of the exponential mapping on $SO(3)$ is given by the Euler-Rodrigues formula, and $\boldsymbol{\psi}(t) \in \mathcal{R}^3$ is the instantaneous rotation vector. Moreover, by following Magnus [12], a solution of (6.35) can be written in the form of (6.38) if $\hat{\boldsymbol{\psi}}(t) \in so(3)$ is a solution of the ODE system in the Lie-algebra

$$\dot{\hat{\boldsymbol{\psi}}} = \mathrm{dexp}^{-1}_{-\hat{\boldsymbol{\psi}}}(\hat{\boldsymbol{\omega}}(\mathbf{Q}(t))), \quad \hat{\boldsymbol{\psi}}_0 = \mathbf{0}, \tag{6.39}$$

and operator $\mathrm{dexp}^{-1}_{-\hat{\boldsymbol{\psi}}}$ is defined by

$$\mathrm{dexp}^{-1}_{-\hat{\boldsymbol{\psi}}}(\hat{\boldsymbol{\omega}}) = \hat{\boldsymbol{\omega}} + \frac{1}{2}\left[\hat{\boldsymbol{\psi}}, \hat{\boldsymbol{\omega}}\right] + \frac{1}{12}\left[\hat{\boldsymbol{\psi}}, \left[\hat{\boldsymbol{\psi}}, \hat{\boldsymbol{\omega}}\right]\right] + \cdots = \sum_{j=0}^{\infty} \frac{B_j}{j!}(-\mathrm{ad}^j_{\hat{\boldsymbol{\psi}}}(\hat{\boldsymbol{\omega}})), \tag{6.40}$$

where the adjoint operator $\mathrm{ad}_{\hat{\boldsymbol{\psi}}}$ is given as Lie-bracket (6.33), and B_j are Bernoulli numbers [12].

6.3.2 *Modified Störmer-Verlet Integration Scheme on SO(3)*

The algorithm for a free-spinning rigid body rotational dynamics with the direct update on $SO(3)$, inspired by the Störmer-Verlet integration scheme in a linear vector space [11], can be written in the form [35]

$$\boldsymbol{\omega}^{n+\frac{1}{2}} = \boldsymbol{\omega}^{n} - \frac{h}{2}(\mathbf{I}^{-1}\hat{\boldsymbol{\omega}}^{n}\mathbf{I}\boldsymbol{\omega}^{n}) \tag{6.41a}$$

$$\mathbf{R}^{n+1} = \mathbf{R}^{n}\mathbf{Q}^{n} = \mathbf{R}^{n}\exp(h\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}}) \tag{6.41b}$$

$$\boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^{n+\frac{1}{2}} - \frac{h}{2}(\mathbf{I}^{-1}\hat{\boldsymbol{\omega}}^{n+1}\mathbf{I}\boldsymbol{\omega}^{n+1}) \tag{6.41c}$$

Here, by following [35], the first order approximation $\hat{\boldsymbol{\psi}} = h\hat{\boldsymbol{\psi}}^{n+\frac{1}{2}} = h\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}}$ is used in (6.38) for the incremental rotation vector and inserted in (6.41b) [while the first order approximation is also used for the discretisation in (6.39)]. The update expressions in Lie algebra (6.41a) and (6.41c) are constructed on the basis of the velocity field discretisation pattern of the original Störmer-Verlet algorithm. As discussed in [35], in order to preserve the system coadjoint orbits, the Eqs. (6.41a) and (6.41c) can be replaced by the expressions derived on the basis of the Eq. (6.34) [or (6.36)]. To this end, we obtain the modified Lie-Störmer-Verlet integration scheme for the forced unconstrained rigid body rotation in the form

$$\boldsymbol{\omega}^{n+\frac{1}{2}} = \mathbf{I}^{-1}(\exp(-\frac{h}{2}\hat{\boldsymbol{\omega}}^{n})(\mathbf{I}\boldsymbol{\omega}^{n} + \frac{h}{2}\mathbf{T}^{n})), \tag{6.42a}$$

$$\mathbf{R}^{n+1} = \mathbf{R}^{n}\exp(h\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}}), \tag{6.42b}$$

$$\begin{aligned}\boldsymbol{\omega}^{n+1} = \mathbf{I}^{-1}(&\exp(-\frac{h}{2}\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}})(\exp(-\frac{h}{2}\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}})(\mathbf{I}\boldsymbol{\omega}^{n} \\ &+ \frac{h}{2}\mathbf{T}^{n}) + \frac{h}{2}\exp(\frac{h}{2}\hat{\boldsymbol{\omega}}^{n+\frac{1}{2}})\mathbf{T}^{n+1})),\end{aligned} \tag{6.42c}$$

where the forcing torque term $\mathbf{T}$ in (6.42c) is introduced at the both ends of the integration interval in order to obtain better conservation properties of the algorithm [35]. By inspection of (6.42), it is clear that for a free spinning body ($\mathbf{T} = 0$) the relation $\mathbf{R}^{n+1}\mathbf{I}\boldsymbol{\omega}^{n+1} = \mathbf{R}^{n}\mathbf{I}\boldsymbol{\omega}^{n}$ is valid. That is, the proposed algorithm exactly preserves spatial angular momentum of a free body. This is confirmed in Fig. 6.6, where it is visible that preservation of the spatial angular momentum $\mathbf{Y} = \mathbf{R}\mathbf{I}\boldsymbol{\omega}$ of a free-spinning body is satisfied and it is independent of the integration step-length.

Technically, Fig. 6.6 shows the convergence of the proposed formulation in the norm $\|\mathbf{Y} - \mathbf{Y}_{\text{converged}}\|_2$, where the reference value $\mathbf{Y}_{\text{converged}} = \mathbf{Y}(t = 1)$ has been computed with a step size of $h = 1e - 5$. The norms of the error are evaluated at step sizes 8, 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024, 1/2048 and 1/4096. The other integration algorithms, whose results are presented in Fig. 6.6, are semi-explicit 2nd order Newmark method written in Lie-group setting [15] (currently one of the best-performing 2nd order geometric algorithms for rigid body rotational dynamics) and 4th order Runge-Kutta Munthe-Kaas method for the ODE integration on Lie groups. The angular velocity initial condition is set to $\boldsymbol{\omega}_0 = \begin{bmatrix} 0.45549 & 0.82623 & 0.03476 \end{bmatrix}^T$ and the rigid body inertia tensor is given by matrix $\mathbf{I} = \text{diag}(0.9144, 1.098, 1.66)$.

The Fig. 6.7 illustrates the convergence of the Lie-Störmer-Verlet scheme for Lagrangian top in the norm of the error in the rotation matrix $\|\mathbf{R} - \mathbf{R}_{\text{converged}}\|_2$

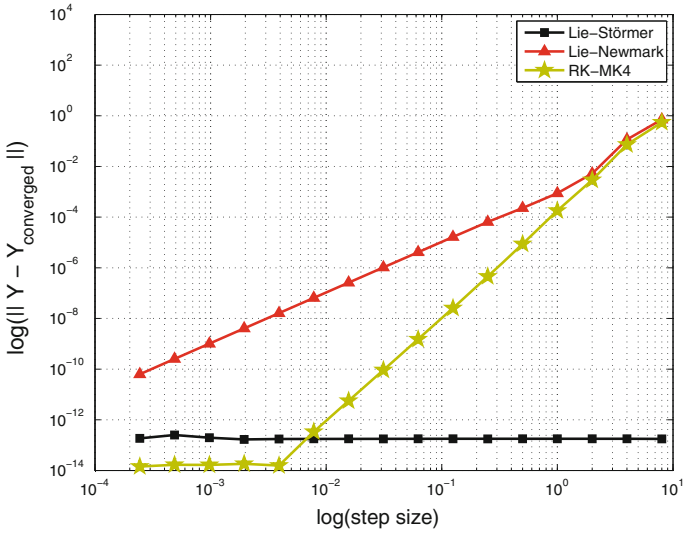


Fig. 6.6 Convergence in the norm of the error in the spatial angular momentum of a free body rotational motion

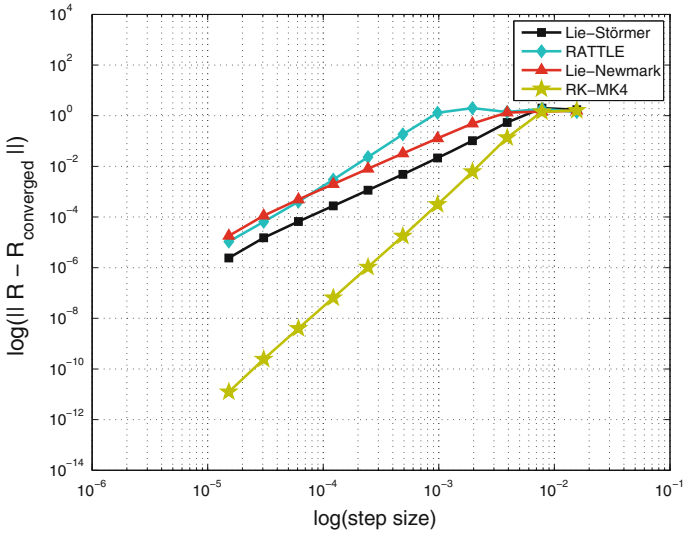


Fig. 6.7 Convergence in the norm of the error in the rotation matrix (*Lagrangian top*)

for decreasing values of the integration step h(1/64, 1/128, 1/256, 1/512, 1/1024, 1/2048, 1/4096, 1/8192, 1/16384, 1/32768 and 1/65536). The reference solution for the rotation matrix $\mathbf{R}_{\text{converged}} = \mathbf{R}(t = 1)$ is computed using the step size $h = 1e - 5$. In this figure, one can see that the proposed formulation exhibits

a second-order convergence (graph 'Lie-Störmer'). Also, the proposed algorithm clearly outperforms the Lie-Newmark and RATTLE algorithms (RATTLE algorithm is 2nd order integration method for the rotational dynamics of a rigid body [17], whose starting point is also Störmer-Verlet scheme), while the RK-MK4 integration scheme, as a 4th order scheme, yields the best accuracy, as expected.

In this example, the configuration space of heavy top is $SO(3)$ [21] and dynamical model is formulated in the classical ODE form on the basis of Euler's rotational equation: dynamical equilibrium of the top rotation around the base point is expressed in terms of $\dot{\boldsymbol{\omega}}$, rotation tensor and gravity force. In the standard units, the inertia tensor with respect to the fixed point has been set as $\mathbf{I}_{\mathrm{FP}} = \mathrm{diag}(15.234375, 0.46875, 15.234375)$, the body mass is set as 15 and the standard gravity acceleration is applied at the local reference point at unity distance along the axis. As it is shown in [35], the presented Lie-Störmer-Verlet method also preserves other integrals of motion of Lagrangian top algorithm in a very satisfactorily manner.

6.3.3 Higher-Order Coadjoint-Preserving Integration Scheme with the Simultaneous Kinematic Reconstruction on SO(3)

The modified Störmer-Verlet scheme in Lie group setting, described in the previous chapter, is a 2nd order integration method. In order to construct the coadjoint orbit preserving integration scheme of a higher order of accuracy, the instantaneous rotation vector $\boldsymbol{\psi}(t) \in \mathcal{R}^3$ in (6.38) should be determined by solving ODE in the Lie algebra (6.39) via higher order integration algorithm. Here, any classical vector space higher order integration method can be used [8, 14].

Furthermore, in addition to an update step at the dynamical level on the basis of (6.34) or (6.36), the same instantaneous rotation vector $\boldsymbol{\psi}(t) \in \mathcal{R}^3$ [that is determined for the step by solving (6.39)] can be also used for the step kinematic reconstruction on $SO(3)$. This stems from the fact that the $SO(3)$ kinematic reconstruction equation

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\hat{\boldsymbol{\omega}}(t), \tag{6.43}$$

has the same mathematical structure as Lie-Poisson's system given by (6.31) or (6.32), meaning that it's update can be written in the same form as (6.34) or (6.36). Indeed, for the n-th step we can write

$$\mathbf{R}_{n+1} = \mathbf{R}_n \mathbf{Q}(t) = \mathbf{R}_n \exp(\hat{\boldsymbol{\psi}}(t)), \quad t \geq t_n, \tag{6.44}$$

where $\hat{\boldsymbol{\psi}}(t) \in so(3)$ is a solution of the ODE system in the Lie-algebra (6.39). This means that for the n-th step, the ODE system (6.39) has to be solved only ones and then both updates at the dynamical and kinematical level can be calculated in a straightforward manner by using (6.34) and (6.44) respectively. This makes the proposed scheme particularly efficient.

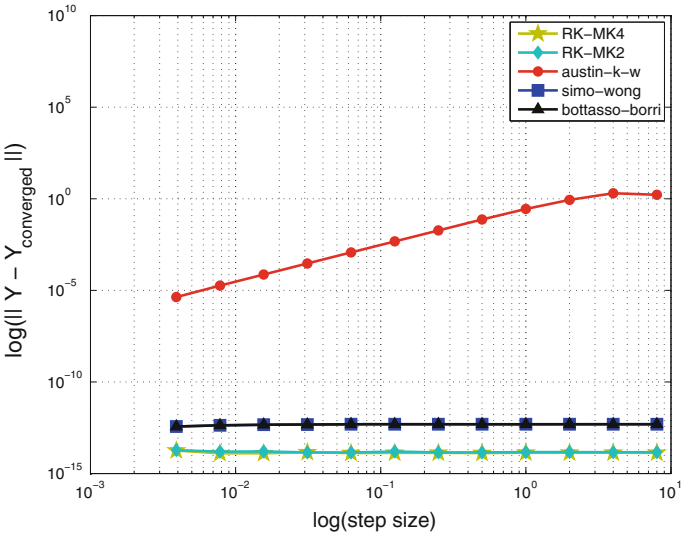


Fig. 6.8 Convergence in the norm of the error in the spatial angular momentum of a free body rotational motion

Since the update at the dynamical level is based on the coadjoint orbit preserving update (6.34), the magnitude of the angular momentum of a free-body in the body-attached frame will be 'exactly' conserved, no matter which integration method is used for solving (6.39). Moreover, by closer inspection of the algorithm, it is clear that for a free spinning body the relation $\mathbf{R}^{n+1}\mathbf{y}^{n+1} = \mathbf{R}^{n}\mathbf{y}^{n}$ is satisfied. That is, the proposed algorithm exactly preserves spatial angular momentum of a free body (similarly as the modified Störmer-Verlet scheme in the previous chapter).

For the integration purposes, in this chapter we have used 4th and 2nd order Runge-Kutta (RK) method i.e. the whole algorithm is based on the Munthe-Kaas method [14, 25] that operates on the Lie-group and uses aforementioned RK algorithm for solving the Lie algebra ODE equation (6.39).

The conservative character of the algorithm is shown in Fig. 6.8, where it is visible that preservation of the spatial angular momentum of a free-spinning body is independent of the integration step-length. Also, as it is expected since this integral of motion is exactly preserved by the proposed method (see above), this is equally true for the both order of the tested accuracies (RK-MK4 and RK-MK2).

The 4th and 2nd order of accuracy (which is dependent on the RK method that is used for solving (6.39), which also, in turn, determines how many terms will be used at RHS in (6.40) for the $\mathrm{dexp}^{-1}_{-\hat{\psi}}$ operator [12]) of the described algorithm are also visible in Fig. 6.9. Here, the both versions are compared within the framework of integration of the free rigid body rotation with the well-known 2nd order geometric algorithms described in the Refs. [1, 3, 32]. Figures 6.8 and 6.9 show the convergence of the proposed formulation in the norms $\left\|\mathbf{Y}-\mathbf{Y}_{\text{converged}}\right\|_2$, and $\left\|\mathbf{R}-\mathbf{R}_{\text{converged}}\right\|_2$ for the

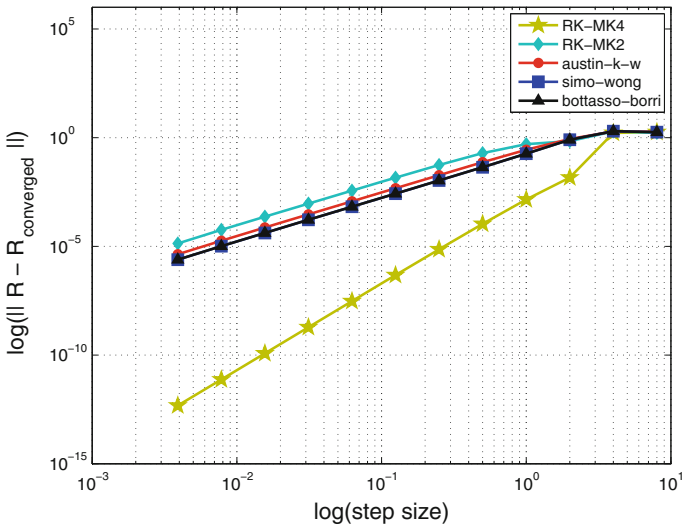


Fig. 6.9 Convergence in the norm of the error in the rotation matrix

freely spinning body, where the spatial angular momentum $\mathbf{Y}_{\text{converged}} = \mathbf{Y}(t = 10)$ and the rotation matrix $\mathbf{R}_{\text{converged}} = \mathbf{R}(t = 10)$ have been computed with a step size of $h = 1e - 3$. The norms of the error are evaluated at step sizes 8, 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128 and 1/256.

6.4 Conclusion

A formulation of the motion equations of the constrained MBS as index 1 DAE system on a Lie group has been presented. It is shown that this system can be solved numerically with the Lie group ODE integration methods following the established MBS methodology. In this chapter, the equations are treated with the MK integration scheme and the Lie-group integration scheme that is based on the Störmer-Verlet algorithm with the direct $SO(3)$ upgrade. This gives rise to a coordinate-free, thus singularity-free, modelling and integration of MBS motion equations.

The Lie-group-setting further provides a framework for the design of geometric integration schemes that preserve coadjoint orbits and conserve integrals of motion. To this end, the Lie-Störmer-Verlet integration scheme is presented. The method is 2nd order accurate and it is angular momentum preserving (it exactly preserves spatial angular momentum of a free body and magnitude of a free-body angular momentum in the body attached frame). Although the method is fully explicit, it generally outperforms two of the best-performing integral-conserving schemes of the 2nd order of accuracy (the semi-explicit Lie-Newmark algorithm and the implicit RATTLE algorithm). The method also performs better than the 4th order explicit RK-MK4

integration algorithm in terms of the conservation of the rigid body motion integrals (free body angular momentum and Lagrangian top motion integrals [35]). The higher-order coadjoint-preserving integration scheme with the simultaneous kinematic reconstruction on $SO(3)$ is discussed in the last part of the chapter. Like the Lie-Störmer-Verlet scheme, this algorithm exactly preserves spatial angular momentum of a free body. It attains a numerically efficient form that makes it easily to be applied to MBS simulations.

Acknowledgments The authors thank Dario Zlatar, Ph.D. student at University of Zagreb, for programming part of the numerical experiments presented in the chapter.

References

1. Austin M, Krishnaprasad PS, Wang LS (1993) Almost Lie-Poisson integrators for the rigid body. J Comput Phys 107:105–117
2. Blajer W (2011) Methods for constraint violation suppression in the numerical simulation of constrained multibody systems—a comparative study. Comput Methods Appl Mech Eng 200:1568–1576
3. Bottasso CL, Borri M (1998) Integrating finite rotations. Comput Methods Appl Mech Eng 164:307–331
4. Brüls O, Cardona A, Arnold M (2012) Lie group generalized-alpha time integration of constrained flexible multibody systems. Mech Mach Theory 48:121–137
5. Brüls O, Cardona A (2010) On the use of Lie group time integrators in multibody dynamics. J Comput Nonlinear Dyn 5(3)
6. Bullo F, Murray RM (1995) Proportional derivative (PD) control on the Euclidean group. CDS technical report 95-010
7. Crouch PE, Grossman R (1993) Numerical integration of ordinary differential equations on manifolds. J Nonlinear Sci 3(1):1–33
8. Engø K, Faltinsen S (1999) Numerical integration of Lie-Poisson systems while preserving coadjoint orbits and energy. Report in Informatics No. 179, University of Bergen
9. Engø K, Marthinsen A (2001) A note on the numerical solution of the heavy top equations. Multibody Syst Dyn 5:387–397
10. Erlicher S, Bonaventura L, Bursi O (2002) The analysis of the generalized- method for nonlinear dynamic problems. Comp Mech 28:83–104
11. Hairer E, Lubich C, Wanner G (2003) Geometric numerical integration illustrated by the Störmer-Verlet method. Acta Numer 12:399–450. doi:10.1017/S0962492902000144
12. Hairer E, Lubich C, Wanner G (2006) Geometric numerical integration. Springer, Berlin
13. Holm Darryl D (2008) Geometric mechanics. Rotating, translating and rolling, Part II. Imperial College Press, London
14. Iserles A, Munthe-Kaas HZ (2000) Lie-group methods. Acta Numer 215–365
15. Krysl P, Endres L (2005) Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies. Int J Numer Meth Eng 62:2154–2177
16. Krysl P (2005) Explicit momentum-conserving integrator for dynamics of rigid bodies approximating the midpoint Lie algorithm. Int J Numer Methods Eng 63(15):2171–2193
17. Leimkuhler B, Reich S (2004) Simulating hamiltonian dynamics. Cambridge University Press, UK
18. Lewis D, Simo JC (1994) Conserving algorithms for the dynamics of hamiltonian systems on Lie groups. J Nonlinear Sci 4:253–299
19. Mäkinen J (2001) Critical study of Newmark-scheme on manifold of finite rotations. Comp Methods Appl Eng 191:817–828

20. Marsden JE, West M (2001) Discrete mechanics and variational integrators. Acta Numer 357–514
21. Marsden JE, Ratiu T (1998) Mechanics and symmetry. Springer, New York
22. Marthinsen A, Munthe-Kaas H, Owren B (1997) Simulation of ordinary differential equations on manifolds—some numerical experiments and verifications. Model Ident Control 18(1):75–88
23. Müller A, Terze Z (2014) On the choice of configuration space for numerical Lie group integration of constrained rigid body systems. Int J Comput Appl Math 262:3–13
24. Müller A, Terze Z (2014) A constraint stabilization method for time integration of constrained multibody systems in lie group setting. In: ASME 2014 international design engineering technical conferences on 9th international conference on multibody systems, 10th nonlinear dynamics, and control (MSNDC), 12–15 Aug 2014, Buffalo, New York
25. Munthe-Kaas H (1998) Runge Kutta methods on Lie groups. BIT 38(1):92–111
26. Munthe-Kaas H (1999) High order Runge-Kutta methods on manifolds. Appl Numer Math 29:115–127
27. Munthe-Kaas H, Owren B (1999) Computations in a free Lie algebra. Phil Trans R Soc A 357:957–981
28. Murray RM, Li Z, Sastry SS (1993) A mathematical Introduction to robotic manipulation. CRC Press, Boca Raton
29. Owren B, Marthinsen A (1999) Runge-Kutta methods adapted to manifolds and based in rigid frames. BIT 39:116–142
30. Park J, Chung WK (2005) Geometric integration on Euclidean group with application to articulated multibody systems. IEEE Trans Rob Automat 21(5):850–863
31. Selig JM (1996) Geometrical methods in robotics. Springer, New York
32. Simo JC, Wong KK (1991) Unconditionally stable algorithms for the orthogonal group that exactly preserve energy and momentum. Int J Numer Methods Eng 31:19–52
33. Terze Z, Müller A, Zlatar D (2012) DAE index 1 formulation for multibody system dynamics in Lie-group setting. In: 2nd Joint international conference on multibody system dynamics (IMSD), 29 May–1 June 2012, Stuttgart
34. Terze Z, Müller A, Zlatar D (2013) Lie-group integration method for constrained multibody systems in state space. Multibody Syst Dyn (submitted)
35. Terze Z, Müller A, Zlatar D (2014) Modified Störmer-Verlet integration scheme for rotational dynamics in Lie-group setting. ASME J Comput Nonlinear Dyn (submitted)
36. Zhanhua M, Rowley CW (2009) Lie-Poisson integrators: a hamiltonian, variational approach. Int J Numer Meth Eng. doi:10.1002/nme.2812

Chapter 7
Solvability of Geometric Integrators for Multi-body Systems

Marin Kobilarov

Abstract This chapter is concerned with the solvability of implicit time-stepping methods for simulating the dynamics of multi-body systems. The standard approach is to select a time-step based on desired level of accuracy and computational efficiency of integration. Implicit methods impose an additional but often overlooked requirement that the resulting nonlinear root-finding problem is solvable and has a unique solution. Motivated by empirically observed integrator failures when using large time-steps this work develops bounds on the chosen time-step which guarantee convergence of the root-finding problem solved with Newton's method. Second-order geometric variational integrators are used as a basis for the numerical scheme due to their favorable numerical behavior. In addition to developing solvability conditions for systems described by local coordinates, this work initiates a similar discussion for Lie group integrators which are a favored choice for floating-base systems such as robotic vehicles or molecular structures.

7.1 Introduction

This work considers the solvability of implicit low-order numerical integrators for multi-body systems with respect to the choice of integration time-step. Our main focus is on geometric variational integrators [1, 15], i.e. integrators which by construction preserve the following physical invariants of the continuous system: symmetries due to conservation laws and associated momentum evolution, configuration space structure such as arising in freely rotating rigid bodies, symplectic phase-space structure. Integrators that respect such variational properties exhibit improved numerics and remedy many practical issues in physically based simulation and animation [2]. In addition, they provide good energy conservation over exponentially

M. Kobilarov (✉)
Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21208, USA
e-mail: marin@jhu.edu

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_7,

long simulation times for non-dissipative systems. When non-conservative forces are present, symplectic structure preservation results in a much-improved treatment of damping that is essentially independent of time step [3]. Our focus on such integrators is also motivated by their successful application to multibody systems [4–10].

Variational integrators could therefore be used as a basis for developing *computationally efficient* algorithms by choosing large time-steps while still retaining desired accuracy. A standard approach is to select the time-step to achieve a desired local or global integration error (e.g. see [11–13] in the context of variational integrators and [14] for the general setting of implicit method). But there is also another key condition that must be satisfied, i.e. the algorithmic solvability of the resulting integrator. This is a key issue since almost all variational time-stepping methods for nonlinear systems are implicit and require the solution of potentially complex nonlinear equations. To the author's knowledge, the issue of implicit integrator *solvability* and its connection to time-step selection has not received enough attention despite the wide use of error-based adaptive time-step selection methods. This turns out to be in fact a central issue for gaining efficiency since, as we show, the success of the numerical root-finding method depends on enforcing *strict bounds on the chosen time-step*. To illustrate this point, consider Fig. 7.1 showing the integration of a simple three-link multi-body system using a symplectic forward Euler method [15]. Based on this empirical evidence there is a clear threshold of the chosen time-step h somewhere in the range $h \in (0.125, 0.5)$ seconds above which the integrator always fails due to divergence of the employed Newton's method. This upper bound could become much lower and thus impose stricter time-step limits as the system increases in complexity.

The main goal if this chapter is therefore to obtain formal bounds on the chosen time-steps h to guarantee solvability of the implicit integrator update. Presently, we obtain such bounds for multi-body systems evolving in a generalized coordinate space $Q = \mathbb{R}^n$ (as described in Sect. 7.4) that *are not* subject to unilateral constraints, e.g. from intermittent contacts or collisions. The second goal of the chapter is to extend those methods to geometric Lie group integrators for floating-base systems with configuration space $Q = SE(3) \times \mathbb{R}^m$, where $m = n - 6$ denotes the number of internal degrees of freedom from movable joints (Sect. 7.5). A number of methods have been developed to take advantage of the differential-geometric and Lie group structure naturally present in multi-body dynamics for numerical integration purposes [16–19]. For our purposes, we are interested in a coordinate-invariant treatment of evolution in the Euclidean group $SE(3)$ to avoid singularities and associated time-step restrictions as well chart switching necessary with coordinates such as Euler angles. The resulting algorithms exhibit surprisingly accurate numerical behavior even at large time-steps. The convergence properties developed for coordinate spaces do not directly apply to Lie groups and need to be considered in a more extensive study. Initial observations for the simplest case of a single rigid body and encouraging results related to time-step regularity conditions are presented in Sect. 7.3.

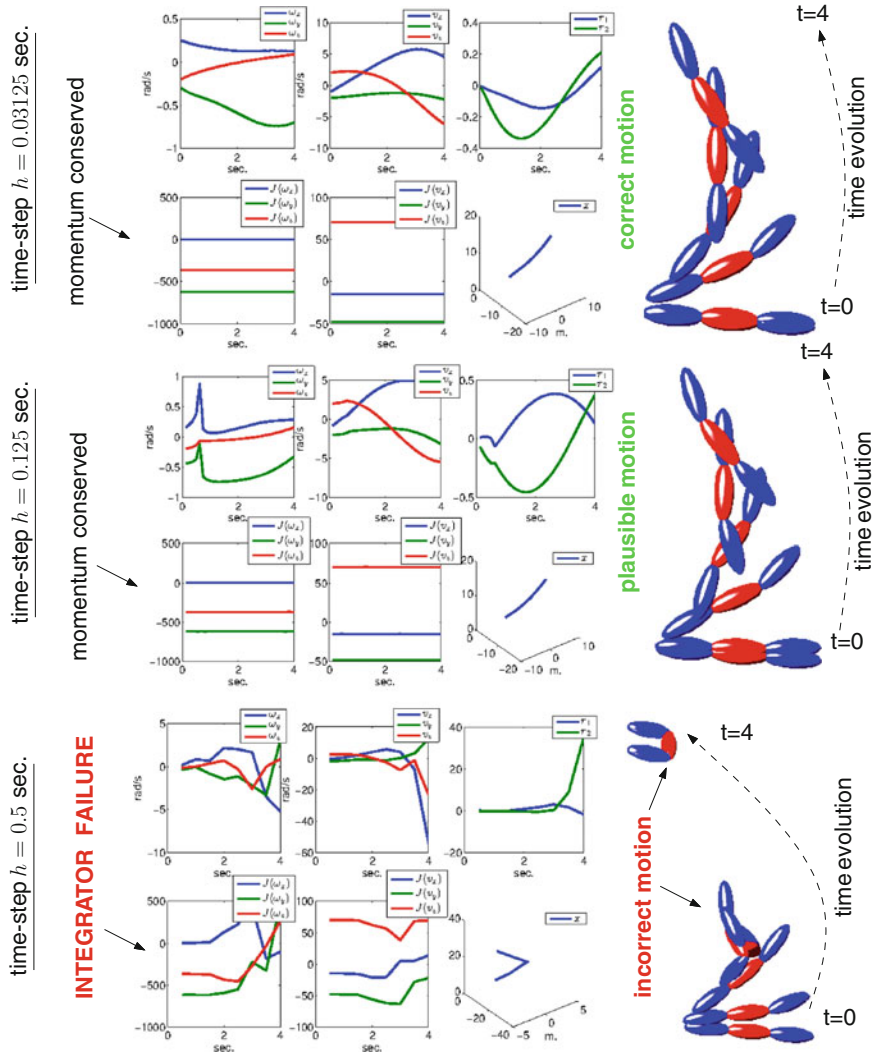


Fig. 7.1 Simulation of a conservative three-link system at three different time-steps. The simulation is qualitatively correct for all time steps below $h = 0.125\,\mathrm{s}$ but suddenly breaks down at higher time-steps. It turns out that this is caused either by crossing singular points of the implicit integrator Jacobian or by divergence of the employed Newton's method. This work seeks a priori conditions based on the dynamical model to find bounds on h avoiding such problems

7.2 Background on Variational Integrators

A mechanical integrator advances a dynamical system forward in time. Such numerical algorithms are typically constructed by directly discretizing the differential equations that describe the trajectory of the system, resulting in an *update rule* to compute

the next state in time. In contrast, variational integrators [1] are based on the idea that the update rule for a discrete mechanical system (i.e., the time stepping scheme) should be derived *directly* from a variational principle rather than from the resulting differential equations. This concept of using a unifying principle from which the equations of motion follow (typically through the calculus of variations [20]) has been favored for decades in physics. Chief among the variational principles of mechanics is *Lagrange D'Alembert's principle* which states that the path $q(t)$ (with endpoint $q(t_0)$ and $q(t_1)$) taken by a mechanical system subject to forces $f(t)$ satisfies the *virtual work* principle $\delta \int_{t_0}^{t_1} L(q, \dot{q})dt + \int_{t_0}^{t_1} f(t)\delta q(t) = 0$, i.e., the state variables $(q, \dot{q})$ evolve such that any variation of the time integral of the *Lagrangian* L of the system (equal to the kinetic minus potential energy) must result from the work done by the force f.

Practically speaking, variational integrators based on Lagrange D'Alembert's principle first approximate the time integral of the continuous Lagrangian and the integral of forces by a *quadrature* rule. This is accomplished using a "discrete Lagrangian," which is a function of two consecutive states q_k and q_{k+1} (corresponding to time t_k and t_{k+1}, respectively):

$$L_d(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))\mathrm{d}t.$$

and "discrete forces" f_d according to

$$f_d^-(q_k, q_{k+1}, u_k, u_{k+1})\delta q_k + f_d^+(q_k, q_{k+1}, u_k, u_{k+1})\delta q_{k+1} \approx \int_{t_k}^{t_{k+1}} f(q(t), \dot{q}(t), u(t))\delta q(t),$$

where the function $f(q, \dot{q}, u)$ defines generalized forces including control inputs u acting on the system and the discrete *left* and *right* forces f_d^- and f_d^+, respectively, approximate the virtual work on the left (resp. right) section of the interval $[t_k, t_{k+1}]$. A discrete variational principle can now be formulated over the whole path $\{q_0, ..., q_N\}$ and control inputs $\{u_0, \ldots, u_N\}$ defined by the successive position at times $t_k = kh$. This discrete principle requires that

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) + \sum_{k=0}^{N-1} [f_d^-(q_k, q_{k+1}, u_k, u_{k+1})\delta q_k + f_d^+(q_k, q_{k+1}, u_k, u_{k+1})\delta q_{k+1}] = 0, \tag{7.1}$$

where variations are taken with respect to each position q_k along the path. Thus, if we use D_i to denote the partial derivative w.r.t the ith variable, we must have

$$D_2L_d(q_{k-1},q_k)+D_1L_d(q_k,q_{k+1})+f_d^+(q_{k-1},q_k,u_{k-1},u_k)\\ +f_d^-(q_k,q_{k+1},u_k,u_{k+1})=0 \tag{7.2}$$

for every three consecutive positions q_{k-1}, q_k, q_{k+1} of the mechanical system. The relationship (7.2) is known as the *discrete Euler-Lagrange (DEL) equation* and *defines* an integration scheme which computes q_{k+1} using the two previous positions q_k and q_{k-1} and given forces u_{k-1}, u_k, u_{k+1}.

Simple Example. Consider a system with continuous, typical Lagrangian of the form $L(q,\dot{q})=\frac{1}{2}\dot{q}^T M\dot{q}-V(q)$ (V being a potential function) and subject to control forces only, i.e. $f(q,\dot{q},u)=u$. Define the discrete Lagrangian using the trapezoidal rule

$$L_d(q_k,q_{k+1})=\frac{h}{2}\left[L\left(q_k,\frac{q_{k+1}-q_k}{h}\right)+L\left(q_{k+1},\frac{q_{k+1}-q_k}{h}\right)\right].$$

with discrete forces defined by

$$f_d^-(q_k,q_{k+1},u_k,u_{k+1})=\frac{h}{2}u_k,\qquad f_d^+(q_k,q_{k+1},u_k,u_{k+1})=\frac{h}{2}u_{k+1}$$

The resulting update equation is:

$$M\frac{q_{k+1}-2q_k+q_{k-1}}{h^2}=u_k-\nabla V(q_k),$$

which is a discrete analog of Newton's law $M\ddot{q}=u-\nabla V(q)$. This example can be easily generalized to systems with configuration-dependent mass matrix $M(q)$ or to systems with constraints leading to variants of the update equation.

7.3 Geometric Integrators for the Rigid Body

We first consider geometric integrators for a single rigid body as one of the simplest mechanical system with nonlinear dynamics. The goal is to illustrate two typical geometric integrators and discuss regularity conditions required for their solvability. These results will then be generalized to multi-body systems.

The standard continuous equations of motion of a controlled rigid body is given by (see e.g. [21])

$$\dot{R}=R\hat{\omega} \tag{7.3}$$

$$\mathbb{J}\dot{\omega}=\mathbb{J}\omega\times\omega+u, \tag{7.4}$$

where $R\in SO(3)$ is the rotation matrix, $\omega\in\mathbb{R}^3$ is the angular velocity, $\mathbb{J}$ is the 3×3 inertia tensor and u are the given control inputs. While it is possible to express the

body rotation using coordinates such as Euler angles a more numerically convenient approach is to perform numerical integration on the configuration manifold directly. For instance, the simplest first-order *Euler method* on $SO(3)$ would take the form

$$R_{k+1} = R_k \exp(h\omega_{k+1}), \tag{7.5}$$

$$\omega_{k+1} = \omega_k + h\mathbb{J}^{-1}(\mathbb{J}\omega_k \times \omega_k + u_k), \tag{7.6}$$

where $\exp : \mathbb{R}^3 \to SO(3)$ is the exponential map defined by

$$\exp(\omega) = \begin{cases} I, & \omega = 0 \\ I + \frac{\sin\|\omega\|}{\|\omega\|}\widehat{\omega} + \frac{1-\cos\|\omega\|}{\|\omega\|^2}\widehat{\omega}^2, & \omega \neq 0 \end{cases}, \tag{7.7}$$

with I denoting the identity matrix and the map $\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$ (with $\mathfrak{so}(3)$ begin the space of 3×3 skew-symmetric matrices) defined by

$$\widehat{\omega} = \begin{bmatrix} 0 & -w_3 & w_3 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{7.8}$$

The integrator (7.5)–(7.6) explicitly updates the next state (R_{k+1}, ω_{k+1}) given the current state (R_k, ω_k). The method is more accurate than a coordinate-based Euler method and does not require coordinate chart switching [22]. Nevertheless, similarly to any other Euler method it is only first-order accurate and has poor numerical stability which becomes especially pronounced at large time-steps h.

7.3.1 Implicit Second-Order Methods

A numerically superior integrator results from implicit second-order formulation, for instance based on trapezoidal or midpoint collocation. As an example, a *trapezoidal collocation* of the dynamics (7.4) will result in the semi-explicit integrator

$$R_{k+1} = R_k \exp(h\omega_{k+1}), \tag{7.9}$$

$$\mathbb{J}(\omega_{k+1} - \omega_k) = \frac{h}{2}(\mathbb{J}\omega_k \times \omega_k + \mathbb{J}\omega_{k+1} \times \omega_{k+1}) + hu_k, \tag{7.10}$$

known as the *trapezoidal Lie-Newmark* (TLN) integrator [15, 23]. The dynamics update (7.10) can be equivalently written as

$$A(h\omega_{k+1})^T\mathbb{J}\omega_{k+1} - A(-h\omega_k)^T\mathbb{J}\omega_k = hu_k$$

where the matrix $A(\omega)$ is defined by

$$A(\omega) = I - \frac{1}{2}\widehat{\omega},$$

and is regarded as the truncated (to first-order) right-trivialized derivative inverse [23, 24] of the exponential map, i.e. $\mathrm{dexp}(w)^{-1} = A(\omega) + O(\|w\|^2)$. In contrast, a very similar method employing the *untruncated* derivative is actually a variational symplectic integrator obtained using a Lie group version of the discrete variational principle (7.1) known as the discrete Euler-Poincare principle (see [1, 23, 25–27]).

An example of such a symplectic Lie group integrator known for its efficiency and ease of implementation [23, 27] is defined by

$$R_{k+1} = R_k \mathrm{cay}(h\omega_{k+1}), \tag{7.11}$$

$$[\mathrm{dcay}^{-1}_{h\omega_{k+1}}]^T \mathbb{J}\omega_{k+1} - [\mathrm{dcay}^{-1}_{-h\omega_k}]^T \mathbb{J}\omega_k = hu_k, \tag{7.12}$$

where the Cayley map $\mathrm{cay}: \mathbb{R}^3 \to SO(3)$ approximates the exponential map and is defined by

$$\mathrm{cay}(\omega) = I + \frac{4}{4 + \|\omega\|^2}\left(\widehat{\omega} + \frac{\widehat{\omega}^2}{2}\right), \tag{7.13}$$

while the right-trivialized tangent inverse is defined by

$$[\mathrm{dcay}^{-1}_{\omega}] = I - \frac{\widehat{\omega}}{2} + \frac{\omega\omega^T}{4}. \tag{7.14}$$

One of the special properties of the symplectic integrator (7.11)–(7.12) is that it preserves the spatial momentum $J_k \approx J(kh)$ given by $J_k = R_k[\mathrm{dcay}^{-1}_{-h\omega_k}]^T \mathbb{J}\omega_k$ in the absence of forces, i.e. when $u_k = 0$.

Time-step selection and solvability. The numerical behavior, preservation properties, and associated backward error analysis of these methods has been established [2, 15, 22, 23, 28]. The resulting favorable numerical behavior permits the use of larger time-steps h while maintaining desired accuracy and stability. But how large can h be? To answer this question we next study regularity conditions of the most common iterative method, i.e. Newton's method, which translate to a maximum time-step selection rule required in order to guarantee solvability of the integrators.

7.3.2 Newton's Method and Time-Step Bounds

Either the collocation or the symplectic methods require the solution of nonlinear discrete dynamics equations, in particular Eqs. (7.10) or (7.12), respectively. This can be formulated as the solution of the nonlinear equations $e_{\mathrm{tln}}(\omega_{k+1}) = 0$ or $e_{\mathrm{symp}}(\omega_{k+1}) = 0$ given by

$$e_{\text{tln}}(\omega) = \left[I + \frac{h}{2}\widehat{\omega} \right] \mathbb{J}\omega - c_k, \tag{7.15}$$

where $c_k \in \mathbb{R}^3$ is given and defined by

$$c_k = A(-h\omega_k)^T \mathbb{J}\omega_k + hu_k,$$

and

$$e_{\text{symp}}(\omega) = \left[I + \frac{h}{2}\widehat{\omega} + \frac{h^2}{4}\omega\omega^T \right] \mathbb{J}\omega - d_k, \tag{7.16}$$

where $d_k \in \mathbb{R}^3$ is given and defined by

$$d_k = [\text{dcay}^{-1}_{-h\omega_k}]^T \mathbb{J}\omega_k + hu_k.$$

While it is possible to apply a number of numerical root-finding methods including polynomial and continuation methods, we focus on Newton-like methods since they generalize to the more complex multi-body setting. Newton's method solves the equation $e(\omega) = 0$ using an initial guess ω which is then iterated according to $w = w - [De(w)]^{-1} e(\omega)$, where $De(\omega)$ is the Jacobian of $e(\omega)$ which must be invertible. The Jacobians of the two methods are given by

$$De_{\text{tln}}(\omega) = \mathbb{J} - \frac{h}{2}\widehat{\mathbb{J}\omega} + \frac{h}{2}\widehat{\omega}\mathbb{J}, \tag{7.17}$$

and

$$De_{\text{symp}}(\omega) = \mathbb{J} - \frac{h}{2}\widehat{\mathbb{J}\omega} + \frac{h}{2}\widehat{\omega}\mathbb{J} + \frac{h}{4}\omega^T \mathbb{J}\omega I + \frac{h}{2}\omega\omega^T \mathbb{J}, \tag{7.18}$$

and are positive definite and invertible at $h = 0$. We next compute the range of time-steps h for which the Jacobians remain invertible. First note that it is not difficult to show[1] that

$$\widehat{\omega}\mathbb{J} \leq \frac{1}{2}\|\omega\|(\sigma_+ - \sigma_-)I,$$

where σ_- and σ_+ are the minimum and maximum eigenvalues of $\mathbb{J}$. Therefore, we have

$$De_{\text{tln}}(\omega) \geq \left[\sigma_- - \frac{h}{4}\|\omega\|(\sigma_+ - \sigma_-) \right] I,$$

or equivalently $De_{\text{tln}}(\omega)$ will be always positive definite and hence invertible if the time-step is chosen according to $0 < h < \bar{h}_{\text{tln}}(\omega)$ where the upper bound is defined by

[1] $A \geq B$ for any matrices $A, B \in \mathbb{R}^{n\times n}$ if and only if $x^T Ax \geq x^T Bx$ for all $x \in \mathbb{R}^n$.

$$\bar{h}_{\text{tln}}(\omega) = \begin{cases} \infty \ \text{(i.e. time-step unrestricted)} & \text{if either } \kappa = 1, \text{ or } \|\omega\| = 0 \\ \frac{4}{(\kappa-1)\|\omega\|} & \text{otherwise,} \end{cases} \tag{7.19}$$

where $\kappa = \frac{\sigma_+}{\sigma_-} \geq 1$ is the *condition number* of $\mathbb{J}$. When $\kappa = 1$ (i.e. a spherical body) or when $\|\omega\| = 0$ we have trivially $De_{\text{tln}} = \mathbb{J}$ and there are no restrictions on the time-step.

The symplectic integrator Jacobian $De_{\text{symp}}(\omega)$ satisfies

$$De_{\text{symp}}(\omega) \geq \left[\sigma_- - \frac{h}{4}\|\omega\|(\sigma_+ - \sigma_-) + h^2\|\omega\|^2\left(\sigma_- - \frac{1}{4}\sigma_+\right)\right] I,$$

and will remain positive definite and invertible all time-steps $0 \leq h < \bar{h}_{\text{symp}}(\omega)$ where the upper bound is defined by

$$\bar{h}_{\text{symp}}(\omega) = \begin{cases} \infty \ \text{(i.e. time-step unrestricted)} & \text{if either } \kappa \leq 4\sqrt{7} - 7, \text{ or } \|\omega\| = 0 \\ \frac{\kappa - 1 - \sqrt{\kappa^2 + 14\kappa - 63}}{(8-2\kappa)\|\omega\|} & \text{otherwise.} \end{cases} \tag{7.20}$$

Figure 7.2 illustrates these bounds by plotting the dependence of $h\|\omega\|$ on the condition number κ. The key point is that the lower the condition number and the lower the velocity magnitude $\|\omega\|$ the higher time-step can be chosen. Note that the symplectic integrator appears to have a wider region of convergence and, unlike collocation, there are no restrictions on the maximum time-step for bodies with condition number $\kappa \approx 3.58$ or smaller. Furthermore, empirically the number of failed solutions is a small fraction of the singular cases for the collocation scheme. In general, though both methods are suitable for very efficient and long-term stable integration as long as the time-step h is chosen to satisfy the respective bounds (7.19) or (7.20).

7.4 Variational Integrators for Mechanical Systems in Generalized Coordinates

We next consider the more general setting of mechanical systems in minimal generalized coordinates, i.e. describing the the system joint angles and pose. The equations of motion of multi-body systems can be derived using a Lagrangian in the typical form

$$L(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} - V(q),$$

where $M(q)$ is a positive-definite mass matrix and V is a potential function. The system is also subject to generalized forces $f(q, \dot{q}, u)$ in the form

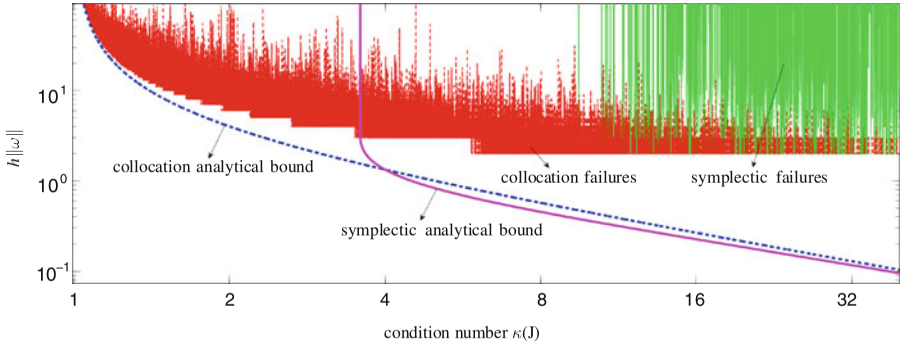


Fig. 7.2 Comparison of maximum time-steps h allowed by the trapezoidal collocation integrator and the symplectic integrator. Both analytical upper bounds are shown as well as empirically computed upper bounds resulting in singular Jacobians (and integrator solution failure) from 50,000 Monte Carlo experiments with varying inertia $\mathbb{J}$ and velocity ω

$$f(q, \dot{q}, u) = f^x(q, \dot{q}) + B(q)u,$$

where f^x encodes any internal and external forces e.g. due to damping or friction and $u \in \mathbb{R}^c$ are the control forces. Our focus is on second-order variational integrators obtained using the discrete Lagrangian

$$L_d(q_k, q_{k+1}) = \frac{h}{2}\left[L(q_k, v_{k+1}) + L(q_{k+1}, v_{k+1})\right],$$

where the discrete velocity $v_k \in \mathbb{R}^n$ is defined by

$$v_k \triangleq \frac{q_k - q_{k-1}}{h},$$

and discrete forces set to

$$f_d^-(q_k, q_{k+1}, u_k, u_{k+1}) = \frac{h}{2} f(q_k, v_{k+1}, u_k),$$
$$f_d^+(q_k, q_{k+1}, u_k, u_{k+1}) = \frac{h}{2} f(q_{k+1}, v_{k+1}, u_{k+1}).$$

This choice of discretization is based on trapezoidal quadrature approximation and as we will show results in a simpler integrator amenable to easier analysis compared to other methods of the same order such as midpoint quadrature (that is not to say that the midpoint rule is inferior).

Discrete Equations of Motion. The general discrete Euler-Lagrange Eq. (7.2) when applied to the mechanical systems result in the implicit integrator

$$\begin{aligned}
&\frac{1}{2}[M(q_k) + M(q_{k+1})]v_{k+1} - \frac{1}{2}[M(q_{k-1}) + M(q_k)]v_k \\
&\quad - \frac{h}{4}\left[(I_n \otimes v_k^T)\nabla M(q_k)v_k + (I_n \otimes v_{k+1}^T)\nabla M(q_k)v_{k+1}\right] + h\nabla V(q_k) \\
&= \frac{h}{2}[f(q_k, v_k, u_k) + f(q_k, v_{k+1}, u_k)]
\end{aligned} \tag{7.21}$$

where the tensor product notation $A \otimes B$ is defined (see e.g. [29]) according to

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{bmatrix}$$

and the matrix gradient ∇M and hence the expression $(I_n \otimes v^T)\nabla M$ are defined as

$$\nabla M = \begin{bmatrix} \frac{\partial M}{\partial q_1} \\ \vdots \\ \frac{\partial M}{\partial q_n} \end{bmatrix}, \qquad (I_n \otimes v^T)\nabla M = \begin{bmatrix} v^T \frac{\partial M}{\partial q_1} \\ \vdots \\ v^T \frac{\partial M}{\partial q_n} \end{bmatrix}.$$

Equivalently, the matrix $(I_n \otimes v^T)\nabla M$ can be expressed in coordinates according to

$$[(I_n \otimes v^T)\nabla M]_{ij} \triangleq \sum_{\ell=1}^{n} \frac{\partial M_{\ell j}}{\partial q_i} v_\ell,$$

where $i, j = 1, \ldots, n$ are the matrix row and column indices, respectively. The relationship (7.21) is expressed more compactly as

$$\begin{aligned} &\frac{1}{2}[M(q_k) + M(q_k + hv_{k+1})]v_{k+1} - \frac{1}{2}[M(q_{k-1}) + M(q_k]v_k + hb_k(v_k, q_k, v_{k+1}) \\ &= hB(q_k)u_k, \end{aligned} \tag{7.22}$$

where the *discrete bias* b_k is defined by

$$\begin{aligned} b_k(v_k, q_k, v_{k+1}) = &-\frac{1}{4}\left[(I_n \otimes v_k^T)\nabla M(q_k)v_k + (I_n \otimes v_{k+1}^T)\nabla M(q_k)v_{k+1}\right] + \nabla V(q_k) \\ &- \frac{1}{2}[f^x(q_k, v_k) + f^x(q_k, v_{k+1})]. \end{aligned} \tag{7.23}$$

The integrator (7.22) can be regarded as the discrete analog of the continuous equations of motion in a standard form (e.g. [30, 31])

$$M(q)\ddot{q} + b(q, \dot{q}) = B(q)u,$$

where the corresponding *continuous bias* term $b(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) - f^x(q, \dot{q})$ encodes Coriolis and centripetal forces defined by the matrix C, gravity forces g, and other forces f^x.

7.4.1 Implicit Time-Stepping Using a Newton Algorithm

The integrator (7.21) is solved in terms of the next velocity v_{k+1} using a numerical root-finding procedure, typically a second-order method such as Newton's method equipped with regularization and line-search procedures. Our goal is to find the root of the equation $e_k(v_{k+1}) = 0$ corresponding to the relationship (7.21) with the mapping $e_k : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$e_k(v) \triangleq \frac{1}{2}[M(q_k) + M(q_k + hv)]v - \frac{1}{2}[M(q_{k-1}) + M(q_k)]v_k + h[b_k(v_k, q_k, v) - B(q_k)u_k]. \tag{7.24}$$

The Jacobian of $e_k(v)$ is

$$De_k(v) = \frac{1}{2}[M(q_k) + M(q_k + hv)] + \frac{h}{2}\left[\nabla M(q_k + hv)^T (I \otimes v) - (I \otimes v^T)\nabla M(q_k) - D_2 f^x(q_k, v)\right], \tag{7.25}$$

where the matrix gradient transpose paired with the tensor product should be understood as

$$\nabla M^T (I \otimes v) \equiv \left[\frac{\partial M^T}{\partial q_1} v, \cdots, \frac{\partial M^T}{\partial q_n} v\right].$$

Note that one of the main practical advantages of using a trapezoidal variational formulation, in addition to its numerical stability, is the relatively simple expression for the Jacobian (7.25). This is not the case if the midpoint rule were used which would involve the unwieldy term $\nabla^2 M$.

Newton's algorithm starts by setting the unknown v to an initial value and iteratively updates it to $v + d$ where the Newton step d is defined by

$$d = -De_k(v)^{-1} e_k(v).$$

We will restrict our analysis to this "pure" version of the algorithm which also employs the previous velocity as an initial value, i.e. the first iteration begins with $v = v_k$. The algorithm is summarized below.

Algorithm 1: $v_{k+1} \leftarrow \texttt{Newton}(v_k, q_k, u_k)$

```
v ← v_k ;
choose time-step h ;
while v has not converged do
    Compute d ∈ ℝ^n such that [De_k(v)]d = −e_k(v) using (7.24) and (7.25);
    v ← v + d
return v
```

We next study the convergence properties of this algorithm. This will be accomplished by assuming certain regularity conditions of the dynamical model and deriving time-step bounds to guarantee convergence.

7.4.2 Convergence of Newton's Method

In order to guarantee solvability of the integrator it is critical to ensure that the time-step h is chosen small enough to ensure convergence of Algorithm 1 or in other words that the true solution can be traced from the initial guess $v = v_k$. This problem has been studied previously [32] for implicit time-stepping methods under general regularity conditions. For mechanical multi-body systems these results need to be extended since it turns out that the Jacobian Lipschitz "constant" normally employed in the Kantorovich-type results [32] is actually a function of the time-step and the current state which requires additional development.

To establish the formal bounds it is necessary to assume the following regularity conditions of the dynamical model:

Assumption 1 Assume that the dynamical system model satisfies the following bounds:

$$m_1(q)I \leq M(q) \leq Im_2(q) \tag{7.26}$$

$$\|M(q) - M(q+w)\| \leq \ell_0(q)\|w\| \tag{7.27}$$

$$\|\nabla M(q)^T (I_n \otimes v)\| \leq \ell_1(q)\|v\| \tag{7.28}$$

$$\left\| [\nabla M(q+w) - \nabla M(q)]^T (I_n \otimes v) \right\| \leq \ell_2(q)\|v\|\|w\| \tag{7.29}$$

$$\|D_2 f^x(q,v)\| \leq \ell_3(q)\|v\| + \ell_4(q) \tag{7.30}$$

$$\|D_2 f^x(q,v) - D_2 f^x(q,v+d)\| \leq [\ell_5(q)\|v\| + \ell_6(q)]\|d\| \tag{7.31}$$

for some known non-negative functions $m_1, m_2, \ell_0, \ell_1, \ldots, \ell_6 \colon Q \to \mathbb{R}_{\geq 0}$ for any $v, w, d \in \mathbb{R}^n$. Furthermore, assume that there is a set $U \subset Q$ and constants $\underline{m}_1, \bar{m}_2, \bar{\ell}_0, \ldots, \bar{\ell}_6 \geq 0$ such that for all $q \in U$:

$$\underline{m}_1 \leq m_1(q), \quad m_2(q) \leq \bar{m}_2, \quad \ell_0(q) \leq \bar{\ell}_0 \quad , \ldots, \quad \ell_6(q) \leq \bar{\ell}_6.$$

The bounds defined in (7.26)–(7.31) are used to obtain regularity conditions which are necessary to guarantee convergence of Newton's method. In particular, the following quantities will be computed in order to construct a convergence proof:

- bound on time-step h to guarantee non-singular Jacobian $De_k(v)$ for a given $v \in \mathbb{R}^n$
- bound on $\|e_k(v_k)\|$ and Jacobian inverse $\|De_k(v_k)^{-1}\|$ at first Newton iteration, i.e. when $v = v_k$
- bound on h so that $De_k(v)$ is invertible for all all Newton iterations staring with $v = v_k$
- Lipschitz bound on the Jacobian $De_k(v)$

This list constitute the steps to be taken in order to construct a Kantorovich-type proof of convergence.

7.4.2.1 Jacobian Regularity

The most basic requirement for convergence is that the Jacobian De_k defined in (7.25) is non-singular, a condition established as follows.

Proposition 1 *Assume that the conditions* (7.26)–(7.31) *hold. The Jacobian* $De_k(v)$ *is non-singular for every time-step* h *such that* $0 \leq h < \bar{h}(q_k, v)$ *where the upper bound is defined by*

$$\bar{h}(q_k, v) = \frac{\sqrt{[\ell_3(q_k)\|v\| + \ell_4(q_k)]^2 + 8\ell_2(q_k)\|v\|^2\underline{m}_1} - \ell_3(q_k)\|v\| - \ell_4(q_k)}{2\ell_2(q_k)\|v\|^2} \tag{7.32}$$

Proof First, note that at $h = 0$ we have $De_k(v) = M(q_k) > 0$, i.e. the Jacobian is positive definite.[2] Therefore, h can be increased as long as De_k remains positive definite and hence invertible. Next we add and subtract the term $\nabla M^T(I_n \otimes v)$ to De_k in (7.25) and since the matrix

$$\nabla M^T(I_n \otimes v) - (I_n \otimes v^T)\nabla M$$

is skew-symmetric it will not affect the Jacobian positivity. Hence we have

$$\begin{aligned} De_k(v) &\geq \frac{1}{2}[M(q_k) + M(q_k + hv)] \\ &\quad + \frac{h}{2}\left[\nabla M(q + hv)^T(I_n \otimes v) - \nabla M(q)^T(I_n \otimes v) - D_2 f^x(q_k, v)\right] \\ &\geq \max\left[m_1(q_k) - \frac{h}{2}\|v\|\ell_0(q_k), \underline{m}_1\right] - \frac{h}{2}\left[\ell_2(q_k)h\|v\|^2 + \ell_3(q_k)\|v\| + \ell_4(q_k)\right], \end{aligned} \tag{7.33}$$

where $\max(\cdot, \cdot)$ takes the maximum of either the local bound at q_k or the global bound $\underline{m}_1$ of the mass matrix. For simplicity, we will employ the global bound so that $De_k(v) > 0$ when h is chosen so that

$$\ell_2(q_k)\|v\|^2h^2 + [\ell_3(q_k)\|v\| + \ell_4(q_k)]h - 2\underline{m}_1 < 0$$

which is satisfied when $h < \bar{h}(q_k, v)$ where $\bar{h}$ is the quadratic equation root defined in (7.32). Note that in case when $\ell_2(q_k)\|v\| = 0$ we have the simpler form

$$\bar{h}(q_k, v) = \frac{2\underline{m}_1}{\ell_3(q_k)\|v\| + \ell_4(q_k)}.$$

Finally, whenever the denominator is zero there are no restriction on the time-step, i.e. $\bar{h}(q_k, v) = \infty$. □

[2] Any matrix (including non-symmetric) $A \in \mathbb{R}^{n\times n}$ is positive definite if $x^TAx > 0$ for all $x \in \mathbb{R}^n$ such that $x \neq 0$.

Linear Damping Forces. Note that the Jacobian regularity bound can be improved when the velocity-dependent terms in the external forces $f^x(q, v)$ are the form $-Kv$ for some matrix $K > 0$. The quadratic condition would then be

$$\ell_2(q_k)\|v\|^2 h^2 + [\ell_4(q_k) - k_1]h - 2\underline{m}_1 < 0,$$

where $k_1 > 0$ is such that $k_1 I \leq K$ with respect to the chosen norm $\|\cdot\|$.

7.4.2.2 Bounding Newton's Method Iterates

Next, we establish bounds on the error function e_k and inverse Jacobian De_k evaluated at the first Newton iteration, i.e. when $v = v_k$. These bound will then be used in computing the the region of convergence of Newton's method determined by the first search step.

Applying the assumption (7.27) twice the following relationship holds

$$\frac{1}{2}[M(q_k + hv_k) - M(q_k - hv_k)]v_k \leq h\ell_0(q_k)\|v_k\|^2,$$

and a bound on the residual e_k evaluated at $v = v_k$ can be established according to

$$\begin{aligned} \|e_k(v_k)\| &\leq h\ell_0(q_k)\|v_k\|^2 + h\|b_k(v_k) - B(q_k)u_k\| \\ &\triangleq hL_0(q_k, v_k). \end{aligned} \tag{7.34}$$

Similarly, the Jacobian satisfies the following bound

$$\begin{aligned} \|De_k(v_k)\| &\leq \bar{m}_2 + h\left[\overline{\ell}_1\|v_k\| + \frac{1}{2}\|D_2 f(q_k, v_k)\|\right] \\ &\triangleq \bar{m}_2 + hL_1(q_k, v_k), \end{aligned} \tag{7.35}$$

while its inverse is bounded according to

$$\|De_k(v_k)^{-1}\| \leq \frac{1}{\underline{m}_1 - hL_1(q_k, v_k)}. \tag{7.36}$$

Note that similarly to (7.33) it is possible to obtain a tighter bound by using $\min[m_2(q_k) + \frac{h}{2}\|v_k\|\ell_0(q_k), \bar{m}_2]$ in place of $\overline{m}_2$ in (7.35) and (7.36) with minimal modification; we employ the simpler $\bar{m}_2$ for clarity.

The first Newton iteration search step $d_0 \in \mathbb{R}^n$ is defined by

$$d_0 \triangleq -[De_k(v_k)]^{-1} e_k(v_k).$$

A convergent Newton algorithm will then perform iterations in the vicinity of the starting value $v = v_k$ in the sense that all iterates v will be contained in the set $\mathcal{B}_0 \triangleq \mathcal{B}(v_k + d_0, \|d_0\|)$, i.e. the ball at point $v_k + d_0$ with radius $\|d_0\|$. More formally, the set $\mathcal{B}(v, r) \subset \mathbb{R}^n$ for a given scalar $r > 0$ is defined as

$$\mathcal{B}(v, r) = \{v + d \mid \|d\| \leq r\}.$$

The following property related to the magnitude of subsequent iterates can be established:

Lemma 7.4.1 *Let assumptions* (7.26)–(7.31) *hold and assume that the time-step h is such that $\mathcal{B}_0 \subset U$. All Newton iterates $v \in \mathbb{R}^n$ are then bounded according to*

$$\|v\| \leq \|v_k\| + \frac{2hL_0(q_k, v_k)}{\underline{m}_1 - hL_1(q_k, v_k)}. \tag{7.37}$$

Proof Assuming the Jacobian $De_k(v)$ is invertible for all $v \in \mathcal{B}_0$, all consequent iterates v will remain inside $\mathcal{B}_0$ which means that

$$\begin{aligned} \|v\| &\leq \|v_k + d_0\| + \|d_0\| \\ &\leq \|v_k\| + 2\|d_0\| \\ &\leq \|v_k\| + 2\|De_k(v_k)^{-1}\|\|e_k(v_k)\|, \end{aligned}$$

and (7.37) follows from (7.34) and (7.36). □

7.4.2.3 The Newton-Kantorovich Condition

So far we obtained condition (7.32) on h guaranteeing that the Jacobian De_k is invertible. The computed upper bound $\overline{h}(q_k, v)$ is a function of v so it is actually not possible to use this bound to guarantee regularity for all Newton iterations a priori, i.e. before executing the algorithm, since obviously v will change at each iteration. But since know that $\|v\|$ is bounded such an a-priori condition is obtained using the upper bound (7.37) by selecting h so that $\phi_k(h) < 0$ where

$$\phi_k(h) = \ell_2 \left[\|v_k\| + \frac{2hL_0}{\underline{m}_1 - hL_1} \right]^2 h^2 + \left[\ell_3 \cdot \left(\|v_k\| + \frac{2hL_0}{\underline{m}_1 - hL_1} \right) + \ell_4 \right] h - \underline{m}_1, \tag{7.38}$$

where all functions ℓ_i and L_i are evaluated at (q_k, v_k). This is now a fourth-order polynomial in h and the upper bound denoted by $\bar{h}_k$ is set to the *smallest positive root of* $\phi_k(h)$.

Next, we establish a Lipschitz condition on the Jacobian. We have, for some $d \in \mathbb{R}^n$:

$$\begin{aligned}\|De_k(v+d) - De_k(v)\| &\leq \frac{1}{2}\|M(q_k + h(v+d)) - M(q_k + hv)\| \\ &+ \frac{h}{2}\Big\|\nabla M\,(q_k + h(v+d))^T\,(I \otimes (v+d)) - \nabla M(q_k + hv)^T (I \otimes v) \\ &- (I \otimes d^T)\nabla M(q_k) - D_2 f^x(q_k, v+d) + D_2 f^x(q_k, v)\Big\| \\ &\leq \frac{h}{2}\left[\bar{\ell}_0 + 2\bar{\ell}_1 + \bar{\ell}_2\|v\| + \ell_5(q_k)\|v\| + \ell_6(q_k)\right]\|d\| \\ &\triangleq h\left[L_2(q_k) + L_3(q_k)\|v\|\right]\|d\|. \end{aligned} \tag{7.39}$$

In order to obtain a Lipschitz bound independent of the velocity v it is necessary to employ bound (7.37) on $\|v\|$ which leads to

$$\begin{aligned}\|De_k(v+d) - De_k(v)\| &\leq h\left[L_2(q_k) + L_3(q_k)\left(\|v_k\| + \frac{2hL_0(q_k, v_k)}{\underline{m}_1 - hL_1(q_k, v_k)}\right)\right]\|d\| \\ &= \frac{h[L_2\underline{m}_1 + L_3\|v_k\|\underline{m}_1] + h^2\,[2L_0L_3 - L_1L_2 - L_1L_3\|v_k\|]}{\underline{m}_1 - hL_1}\|d\| \\ &\triangleq \frac{hL_4 + h^2L_5}{\underline{m}_1 - hL_1}\|d\|, \end{aligned} \tag{7.40}$$

so that the factor in front of $\|d\|$ can now be regarded as the required Lipschitz constant of the Jacobian [32].

Choosing $h < \bar{h}_k$ where $\bar{h}_k$ is the smallest positive root of $\phi_k(h)$ guarantees that the Jacobian is invertible for all Newton iterations assuming the method was initialized with $v = v_k$. This bound can now be combined with the actual sufficient condition for convergence. As a result, a stricter bound will be obtained that is sufficient to guarantee a successful solution.

Proposition 2 *Assume that conditions (7.26)–(7.31) hold and that the time-step h is such that $\phi_k(h') < 0$ for all $h' \in [0, h]$ or equivalently assume that $h < \bar{h}_k$. Furthermore, assume that $\mathcal{B}_0 \subset U$ for all $h \leq \bar{h}_k$. Newton's algorithm then converges super-linearly to a unique solution inside $\mathcal{B}_0$ if the time-step h is chosen so that $\psi(h') < 0$ for all $h' \in [0, h]$ where*

$$\psi(h) = h^3(2L_0L_5 + L_1^3) + h^2(2L_0L_4 - 3\underline{m}_1L_1^2) + h(2\underline{m}_1^2L_1) - \underline{m}_1^3. \tag{7.41}$$

Equivalently, convergence is ensured if the upper bound on h is set to the smallest positive root of $\psi(h) = 0$.

Proof To ensure convergence, the Newton-Kantorovich theorem [33] requires that

$$\frac{hL_4 + h^2L_5}{\underline{m}_1 - hL_1}\|De_k^{-1}(v_k)\|\|d_0\| \leq \frac{1}{2}, \tag{7.42}$$

where the first term corresponds to the Jacobian Lipschitz term (7.40). Note that

$$\begin{aligned}\|De_k^{-1}(v_k)\|\|d_0\| &\leq \|De_k^{-1}(v_k)\|^2\|e_k(v_k)\| \\ &\leq \frac{hL_0(q_k, v_k)}{[\underline{m}_1 - hL_1(q_k, v_k)]^2},\end{aligned}$$

using the computed bounds (7.34) and (7.36), which is then substituted into (7.42) to obtain $\psi(h) < 0$. Since this condition is trivially satisfied for $h = 0$ the upper bound is the smallest positive h for which $\psi(h) = 0$. □

7.5 Variational Lie Group Integrators for Multibody Systems

The integrators developed in Sect. 7.4 are based on generalized coordinates q in the Euclidean space $\mathbb{R}^n$. The configuration space Q is actually only locally isomorphic to $\mathbb{R}^n$ in the sense that any choice of rotational coordinates such as Euler angles cannot globally cover the space of rotations using a single chart. Most floating-base multi-body systems have a configuration space $Q = SE(3) \times \mathbb{R}^m$ with $q = (g, r)$ where $g \in SE(3)$ is the pose of a chosen *base body* and $r \in \mathbb{R}^m$ are the joint angles or *shape variables*. Such representation is sufficient for tree-topology multi-body systems with m internal (i.e. from movable joints) degrees of freedom. A more general graph-topology system with loops is modeled by selecting a spanning tree and enforcing loop constraints using additional multiplier variables. Our goal is to develop geometric variational integrators for such systems which evolve intrinsically on the configuration space Q. These integrators can be regarded as an extension of the single rigid body integrators described in Sect. 7.3 to general multi-body systems. We first focus on the standard continuous setting and then develop the corresponding geometric structure-preserving integrators.

7.5.1 The Continuous Setting

The configuration of a tree-topology multi-body system is defined as $q = (g, r) \in SE(3) \times \mathbb{R}^m$ with velocity given by $v = (\xi, \dot{r}) \in \mathbb{R}^{6+m}$, where $g \in SE(3)$ is the 4×4 pose matrix describing the base body orientation $R \in SO(3)$ and position $x \in \mathbb{R}^3$ according to

$$g = \begin{bmatrix} R & x \\ 0_{1\times 3} & 1 \end{bmatrix}, \qquad g^{-1} = \begin{bmatrix} R^T & -R^T x \\ \mathbf{0} & 1 \end{bmatrix}.$$

and where $\xi = (\omega, \mathsf{v}) \in \mathbb{R}^6$ defines its body-fixed angular velocity $\omega \in \mathbb{R}^3$ and linear velocity $\mathsf{v} \in \mathbb{R}^3$. The body-fixed velocity ξ is related to the configuration using the relationship [21, 31]

$$\dot{g} = g\widehat{\xi},$$

where the operator $\widehat{\cdot}: \mathbb{R}^6 \to \mathfrak{se}(3)$ turns velocities $\xi = (\mathsf{v}, \omega)$ into the 4×4 matrices

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & \mathsf{v} \\ 0_{1\times 3} & 0 \end{bmatrix}. \tag{7.43}$$

The Lagrangian of the system is defined by

$$L(g, r, \xi, \dot{r}) = \frac{1}{2}(\xi, \dot{r})^T \mathcal{M}(r)(\xi, \dot{r}) - V(g, r), \tag{7.44}$$

or more compactly as $L(q, v) = \frac{1}{2} v^T \mathcal{M}(r) v - V(q)$, where the mass matrix $\mathcal{M}$ is defined by (e.g. see [31, 34])

$$\mathcal{M}(r) = \left[\begin{array}{c|c} \mathbb{I}_0 + \sum_{i=1}^n A_i^T \mathbb{I}_i A_i & \sum_{i=1}^n A_i^T \mathbb{I}_i J_i \\ \hline \sum_{i=1}^n J_i^T \mathbb{I}_i A_i & \sum_{i=1}^n J_i^T \mathbb{I}_i J_i \end{array}\right] \tag{7.45}$$

where $\mathbb{I}_i$ is the 6×6 inertia matrix of body i and the adjoint notation $A_i \triangleq \mathrm{Ad}_{g_{0i}^{-1}(r)}$, and Jacobian $J_i \triangleq \sum_{j=1}^n [g_{0i}^{-1}(r) \partial_{r_j} g_{0i}(r)]^\vee$ were employed (the operator $\cdot^\vee$ is the inverse of the operator $\widehat{\cdot}$ defined in (7.43)). Here we use the standard notation $g_{0i}: \mathbb{R}^m \to SE(3)$ to define the transformation between the base body (with index #0) and body #i (see e.g. [31]). The adjoint map Ad_g is defined by the 6×6 matrix

$$\mathrm{Ad}_g = \begin{bmatrix} R & 0 \\ \widehat{x} R & R \end{bmatrix}. \tag{7.46}$$

Various efficient methods exist [30, 34] to compute the Jacobians and the mass matrix recursively exploiting the tree structure of the multi-body system. Finally, assume that the system is subject to generalized forces expressed through the known function $f(q, v, u)$. The variational principle used to obtain the dynamics is

$$\delta \int L(g, r, \xi, \dot{r}) dt + \int \langle f(g, r, \xi, \dot{r}, u), (\eta, \delta r) \rangle = 0, \tag{7.47}$$

where the *left-trivialized* variation $\eta \in \mathbb{R}^6$ is defined by $\eta(t) = (g(t)^{-1} \delta g(t))^\vee$. The resulting equations of motion are obtained by taking variations $(\delta g, \delta r)$ and $(\delta \xi, \delta r)$ subject to the constraint (see e.g. [21])

$$\delta \xi = \dot{\eta} + \mathrm{ad}_\xi \eta,$$

where the adjoint operator ad_ξ is defined by the 6×6 matrix

$$\mathrm{ad}_\xi = \begin{bmatrix} \widehat{\omega} & 0 \\ \widehat{\mathsf{v}} & \widehat{\omega} \end{bmatrix}.$$

This variational constraint stems from the kinematic constraints $\xi = (g^{-1}\dot{g})^{\vee}$ between ξ and g.

Before stating the equations of motion it is necessary to define a procedure for differentiating functions on the Lie group $SE(3)$. This will be accomplished by applying a trivialized gradient as opposed to the standard gradient on $\mathbb{R}^n$ as follows.

Definition 7.5.1 The left-trivialized gradient $g^*\nabla_g V(g) \in \mathbb{R}^6$ of a function $V: SE(3) \to \mathbb{R}$,

$$g^*\nabla V(g) = \nabla_\xi \Big|_{\xi=0} V(g\ \exp(\xi))$$

or in coordinates using the standard basis $\{e_1, \ldots, e_6\}$ of $\mathbb{R}^6$ by

$$g^*\nabla V(g) = \left[\frac{\partial V}{\partial s}\Big|_{s=0}(x\ \exp(se_1)),\ \cdots,\ \frac{\partial V}{\partial s}\Big|_{s=0}(x\ \exp(se_6)) \right]^T.$$

Continuous Equations of Motion. Employing the momenta $\mu = \partial_\xi L$ and $p = \partial_{\dot{r}} L$ the resulting dynamics can be expressed as:

$$\begin{bmatrix} \xi \\ \dot{r} \end{bmatrix} = M(r)^{-1} \begin{bmatrix} \mu \\ p \end{bmatrix} \tag{7.48}$$

$$\begin{bmatrix} \dot{\mu} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} (ad_\xi)^T \mu \\ \frac{1}{2}(I_n \otimes v^T)\nabla M(r)v \end{bmatrix} - \begin{bmatrix} g^*\nabla_g V \\ \nabla_r V \end{bmatrix} + f(q, v, u), \tag{7.49}$$

The system evolution is then fully determined by adding the reconstruction equations

$$\dot{g} = g\widehat{\xi}$$

which corresponds to setting $\dot{R} = R\omega$ and $\dot{x} = R\mathsf{v}$. Note that Eqs. (7.48) and (7.49) can be regarded as an extension of the standard Hamiltonian form of the equations of motion (see e.g. [29]) to floating-base systems. In order to derive the corresponding geometric integrator we next specify a methodology for performing discrete-time updates on Lie groups, such as $SE(3)$, without resorting to local coordinates.

7.5.2 Trajectory Discretization on Lie groups

A trajectory is represented numerically using a set of $N+1$ equally spaced in time points denoted $g_{0:N} := \{g_0, ..., g_N\}$, where $g_k \approx g(kh) \in G$ and $h = T/N$ denotes the time-step. The section between each pair of points g_k and g_{k+1} is interpolated by a short curve that must lie on the manifold (Fig. 7.3). The simplest way to construct such a curve is through a map $\tau: \mathfrak{g} \to G$ and velocity vector $\xi_k \in \mathfrak{g}$ such that $\xi_k = \tau^{-1}(g_k^{-1}g_{k+1})/h$. Here $\mathfrak{g} \equiv T_eG$ denotes the Lie algebra of G. The map is defined as follows.

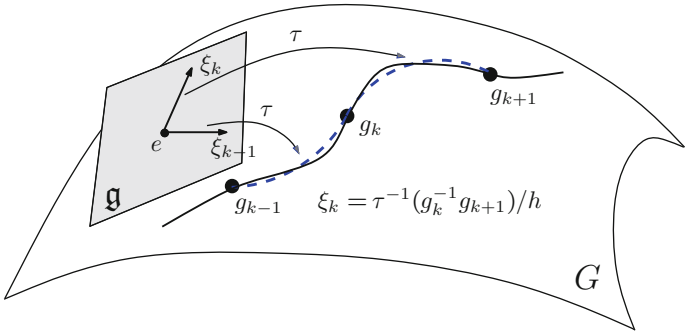


Fig. 7.3 A trajectory (*solid*) on the Lie group G discretized using a sequence of arcs (*dashed*) represented by Lie algebra vectors $\xi_k \in \mathfrak{g}$ through the retraction map τ [27]

Definition 7.5.2 The *retraction map* $\tau : \mathfrak{g} \to G$ is a C^2-diffeomorphism around the origin such that $\tau(0) = e$. It is used to express small discrete changes in the group configuration through unique Lie algebra elements. For our purposes, we consider maps such that $\tau(\xi) = \exp(\xi) + O(\|\xi\|^3)$.

Thus, if ξ_k were regarded as an average velocity between g_k and g_{k+1} then τ is an approximation (to at least second-order) to the integral flow of the dynamics. The important point, from a numerical point of view, is that the difference $g_k^{-1} g_{k+1} \in G$, which is an element of a nonlinear space, can now be represented uniquely by the vector ξ_k in order to enable unconstrained optimization in the linear space $\mathfrak{g}$ for optimal control purposes.

Next, we define the following operators related to τ.

Definition 7.5.3 [23, 28] Given a map $\tau : \mathfrak{g} \to G$, its *right-trivialized tangent* $\mathrm{d}\,\tau_\xi : \mathfrak{g} \to \mathfrak{g}$ and its *inverse* $\mathrm{d}\,\tau_\xi^{-1} : \mathfrak{g} \to \mathfrak{g}$ are such that, for a some $g = \tau(\xi) \in G$ and $\eta \in \mathfrak{g}$, the following holds

$$\partial_\xi \tau(\xi) \cdot \eta = \mathrm{d}\,\tau_\xi \cdot \eta \cdot \tau(\xi), \tag{7.50}$$

$$\partial_\xi \tau^{-1}(g) \cdot \eta = \mathrm{d}\,\tau_\xi^{-1} \cdot (\eta \cdot \tau(-\xi)) \,. \tag{7.51}$$

Note that it can be shown by differentiating the expression $\tau^{-1}(\tau(\xi)) = \xi$ that

$$d\tau_\xi^{-1} \cdot d\tau_\xi \cdot \eta = \eta,$$

to confirm that these linear maps are indeed the inverse of each other.

7.5.3 Retraction Map (τ) Choices

(a) The *exponential map*
$\exp : \mathfrak{g} \to G$, defined by $\exp(\xi) = \gamma(1)$, with $\gamma : \mathbb{R} \to G$ is the integral curve through the identity of the vector field associated with $\xi \in \mathfrak{g}$ (hence, with $\dot{\gamma}(0) = \xi$). The right-trivialized derivative of the map exp and its inverse are defined as

$$\mathrm{dexp}_x\, y = \sum_{j=0}^{\infty} \frac{1}{(j+1)!}\, \mathrm{ad}_x^j\, y, \tag{7.52a}$$

$$\mathrm{dexp}_x^{-1}\, y = \sum_{j=0}^{\infty} \frac{B_j}{j!}\, \mathrm{ad}_x^j\, y, \tag{7.52b}$$

where B_j are the Bernoulli numbers. Typically, these expressions are truncated in order to achieve a desired order of accuracy. The first few Bernoulli numbers are $B_0 = 1$, $B_1 = -1/2$, $B_2 = 1/6$, $B_3 = 0$ (see [15]).

(b) The *Cayley map* $\mathrm{cay} : \mathfrak{g} \to G$ is defined by $\mathrm{cay}(\xi) = (\mathrm{I} - \xi/2)^{-1}(\mathrm{I} + \xi/2)$ and is valid for a general class for quadratic groups that include the groups of interest in the chapter. Based on this simple form, the derivative maps become ([15], §IV.8.3)

$$\mathrm{dcay}_x\, y = \left(e - \frac{x}{2}\right)^{-1} y \left(e + \frac{x}{2}\right)^{-1}, \tag{7.53a}$$

$$\mathrm{dcay}_x^{-1}\, y = \left(e - \frac{x}{2}\right) y \left(e + \frac{x}{2}\right). \tag{7.53b}$$

The third choice is to use canonical coordinates of the second kind (ccsk) [15] which are based on the exponential map and are not considered in this chapter. In our implementation we employ the Cayley map the details for which are given next.

7.5.3.1 The Cayley Map for Rigid Body Transformations

The algorithms developed in this chapter are based on the the Cayley map for $SE(3)$ since it is often a better alternative to the exponential for computational efficiency and ease of implementation that does not require special numerical handling at the origin. With a slight abuse of notation, i.e. assuming the identification $\mathfrak{g} \sim \mathbb{R}^6$, the Cayley map $\tau : \mathbb{R}^6 \to SE(3)$ is defined as (see [27])

$$\tau(\xi) = \begin{bmatrix} I_3 + \frac{4}{4+\|\omega\|^2}\left(\widehat{\omega} + \frac{\widehat{\omega}^2}{2}\right) & \frac{2}{4+\|\omega\|^2}(2I_3 + \widehat{\omega})\, v \\ 0 & 1 \end{bmatrix}, \tag{7.54}$$

while the matrix representation of the right-trivialized tangent inverse $\mathrm{d}\,\tau_\xi^{-1} : \mathbb{R}^6 \to \mathbb{R}^6$ becomes

$$[\mathrm{d}\,\tau_\xi^{-1}] = \begin{bmatrix} I_3 - \frac{1}{2}\widehat{\omega} + \frac{1}{4}\omega\omega^T & 0_3 \\ -\frac{1}{2}\left(I_3 - \frac{1}{2}\widehat{\omega}\right)\widehat{v} & I_3 - \frac{1}{2}\widehat{\omega} \end{bmatrix}. \tag{7.55}$$

7.5.4 Discrete Variational Formulation

With a discrete trajectory in place we follow the approach taken in [25, 26, 35] in order to construct a structure-preserving (i.e. group, momentum, and symplectic) integrator of the dynamics. We make a simple extension to include potential and control forces through a trapezoidal quadrature approximation. In particular, the action in (7.1) is approximated along each discrete segment between points (g_k, r_k) and (g_{k+1}, r_{k+1}) through

$$\int_{kh}^{(k+1)h} L(g, r, \xi, \dot{r})\,\mathrm{d}t \approx h L_d(g_k, g_{k+1}, r_k, r_{k+1}), \tag{7.56a}$$

$$\int_{kh}^{(k+1)h} \langle f, (\eta, \delta r)\rangle dt \approx \left[\langle f_d^-(g_k, g_{k+1}, r_k, r_{k+1}), (\eta_k, \delta r_k)\rangle + \langle f_d^-(g_k, g_{k+1}, r_k, r_{k+1}), (\eta_{k+1}, \delta r_{k+1})\rangle\right]. \tag{7.56b}$$

where the discrete variation $\eta_k \in \mathbb{R}^6$ is defined by $\eta_k = (g_k^{-1}\delta g_k)^\vee$. The discrete Lagrangian and forces are defined by

$$L_d(g_k, g_{k+1}, r_k, r_{k+1}) = \frac{h}{2}\left[L(g_k, r_k, \xi_{k+1}, \Delta r_{k+1}) + L(g_{k+1}, r_{k+1}, \xi_{k+1}, \Delta r_{k+1})\right], \tag{7.57}$$

$$f_d^-(g_k, g_{k+1}, r_k, r_{k+1}, u_k, u_{k+1}) = \frac{h}{2} S(h\xi_{k+1})^T f(g_k, r_k, \xi_{k+1}, \Delta r_{k+1}, u_k), \tag{7.58}$$

$$f_d^+(g_k, g_{k+1}, r_k, r_{k+1}, u_k, u_{k+1}) = \frac{h}{2} S(-h\xi_{k+1})^T f(g_{k+1}, r_{k+1}, \xi_{k+1}, \Delta r_{k+1}, u_{k+1}), \tag{7.59}$$

where the discrete velocities $\xi_k \in \mathbb{R}^6$ and $\Delta r_k \in \mathbb{R}^m$ are defined by

$$\xi_k \triangleq \tau^{-1}(g_{k-1}^{-1} g_k)/h, \qquad \Delta r_k \triangleq (r_k - r_{k-1})/h.$$

The matrix $S(\xi)$ is defined by

$$S(\xi) = \begin{bmatrix} \mathrm{d}\tau_\xi^{-1} & 0 \\ 0 & I_m \end{bmatrix} \tag{7.60}$$

and is interpreted as a Jacobian (or push-forward) map which transforms average vectors along a segment generated by ξ to vectors defined at the beginning of the segment [2, 9]. The map τ and its tangent $\mathrm{d}\tau^{-1}$ are defined in (7.54) and (7.55) and implemented through simple matrix-vector products. Finally, note that we have the following variational constraint which may be obtained through differentiation and application of (7.51),

$$\delta\xi_k = \delta\tau^{-1}(g_{k-1}^{-1}g_k)/h = [-\,\mathrm{d}\,\tau_{h\xi_k}^{-1}\eta_{k-1} + \mathrm{d}\,\tau_{-h\xi_k}^{-1}\eta_k]/h, \tag{7.61}$$

which serves as the basis for applying the variational principle on Lie groups and also the reason why $S(\xi)$ appears in (7.58) and (7.59).

Discrete Equations of Motion. The resulting geometric integrator from applying the principle (7.1) using the discrete Lagrangian (7.57) and forces (7.58) and (7.59) subject to the constraint (7.61) is:

$$g_{k+1} = g_k\tau(h\xi_{k+1}) \tag{7.62}$$

$$r_{k+1} = r_k + h\Delta r_{k+1} \tag{7.63}$$

$$\begin{aligned}\frac{1}{2}S(h\xi_{k+1})^T[M(r_k) + M(r_k + h\Delta r_{k+1})]v_{k+1} - \frac{1}{2}S(-h\xi_k)^T[M(r_{k-1}) + M(r_k)]v_k = \\ \frac{h}{4}\begin{bmatrix} 0 \\ (I_n \otimes v_{k+1}^T)\nabla\mathcal{M}(r_k)v_{k+1} + (I_n \otimes v_k^T)\nabla\mathcal{M}(r_k)v_k \end{bmatrix} - h\begin{bmatrix} g^*\nabla_g V(q_k) \\ \nabla_r V(q_k)\end{bmatrix} \\ + \frac{h}{2}\Big[S(h\xi_{k+1})^T f(q_k, v_{k+1}, u_k) + S(-h\xi_k)^T f(q_k, v_k, u_k)\Big].\end{aligned} \tag{7.64}$$

Applying Newton's Algorithm. The discrete equations of motion (7.62)–(7.64) are used to update the current state $(q_k, v_k) = (g_k, r_k, \xi_k, \Delta r_k)$ to obtain the next state $(q_{k+1}, v_{k+1}) = (g_{k+1}, r_{k+1}, \xi_{k+1}, \Delta r_{k+1})$. This is accomplished by first solving the dynamics (7.64) using a root-finding algorithm such as Newton's method in terms of the unknowns $v_{k+1} = (\xi_{k+1}, \Delta r_{k+1})$ which are then used in the explicit Eqs. (7.62)–(7.63) to obtain the next configuration $q_{k+1} = (g_{k+1}, r_{k+1})$.

7.5.5 Preservation Properties

One of the main benefits of employing the variational numerical framework lies in its preservation properties, summarized as follows.

Symplectic structure. The discrete flow (7.64) preserves the discrete symplectic form, expressed in coordinates as

$$\omega_L = \frac{\partial^2 L_d(q_k, q_{k+1})}{\partial q_k^i \partial q_{k+1}^j} \, \mathrm{d}q_k^i \wedge \mathrm{d}q_{k+1}^j,$$

where $\wedge$ is the standard wedge product between differential forms [21]. The symplectic form is physically related to the phase space structure. Its preservation during integration, for instance, signifies that a volume of initial conditions would not be spuriously inflated or deflated due to numerical approximations. Volume preservation means that the orbits of the dynamics will have a predictable character and no artificial damping normally employed by Runge-Kutta methods is needed to stabilize the system [1].

Momentum Conservation. The discrete dynamics (7.64) also exactly preserves any Lagrangian symmetries. In particular, assume that there is a group G whose action on Q leaves the Lagrangian invariant in the sense that

$$L(q, v) = L(\exp(s\rho)q, v),$$

which implies that

$$L_d(q_k, q_{k+1}) = L_d(\exp(s\rho)q_k, \exp(s\rho)q_{k+1}),$$

for some $\rho \in \mathfrak{g}$, where $\mathfrak{g}$ is the Lie algebra of G, and s is a scalar. In this case the momentum map $J(q_k, q_{k+1}) \cdot \rho = D_1 L_d(q_k, q_{k+1}) \cdot \rho_Q(q_k)$ is preserved [1] where ρ_Q is the infinitesimal generator of the group [21]. Practically speaking, whenever the continuous system preserves momentum, so does the discrete. Any change in the momentum then exactly reflects the work done by non-conservative forces. Such a momentum-symplectic scheme also exhibits long-term stable energy behavior close to the true system energy [1].

For instance, assume that the Lagrangian of a multi-body system is invariant with respect to spatial rotations and translations and that there are no external or control forces acting on the base body. In this case we have $G = SE(3)$ and the momentum map

$$J(q_k, q_{k+1}) = \begin{bmatrix} \mathrm{Ad}^T_{g_k^{-1}} & 0_m \end{bmatrix} \frac{1}{2} S(h\xi_{k+1})^T [M(r_k) + M(r_k + h\Delta r_{k+1})] v_{k+1} \quad (7.65)$$

is exactly preserved by the integrator.

Order of Accuracy. The order of accuracy of the dynamics update depends on the accuracy of the Lagrangian approximation. Since the trapezoidal approximations (7.56a) and (7.56b) are employed then it can be shown (see [1]) that the discrete Eq. (7.64) are at least *second order accurate*. The trapezoidal rule was chosen since it provides one of the simplest second-order scheme. Higher-order methods by proper choice of the Lagrangian, termed symplectic Runge-Kutta (see [15, 23, 26]), are possible but not considered in this work.

Group structure. Finally, the group structure is exactly preserved since each configuration g_k is reconstructed from the previous pose g_{k-1} and the discrete velocity ξ_k using the map τ which by definition maps to the group $SE(3)$. This avoids issues with dissipation and numerical drift associated with reprojection used for instance in explicit methods based on matrix orthogonality constraints or quaternions.

7.5.6 Numerical Example

These numerical properties are illustrated with a simple multibody system consisting of three bodies in 3-D arranged in a chain connected with hinge joints (Fig. 7.4). The system is free-floating with the central body taken as the base body with index #0. No control or external forces are applied in order to verify the integrator momentum conservation properties. Figure 7.5 illustrates the resulting time-histories of the velocities ω and v, joint angles r, momentum components J (corresponding to the vector (7.65)) and position x. The true trajectory was constructed using an Euler step with step-size $h = 0.001\,\mathrm{s}$. while the step-size for both the symplectic method (symp) and Euler method (euler) were $h = 0.1\,\mathrm{s}$. The figures show that momentum is exactly preserved by the symplectic method. The purpose of this study is not to preform detailed comparisons but only to validate the basic numerical features of the method. The main point is that these results motivate a further study to extend the coordinate-based convergence conditions (7.41) to Lie group methods for mechanical systems.

7.6 Conclusion

This chapter considered numerical properties of geometric integrators for multibody systems related to the choice of time-step h. Such methods exhibit favorable numerical stability and accuracy but require the solution of a potentially complex system of nonlinear equations. We showed that the solvability of this system can be guaranteed by ensuring that h is chosen below an upper bound $\bar{h}$ determined from the dynamical model parameters and previous state of the system. The availability of such a bound a priori is important since it could enable predictable computation times for real-time integration or optimal control purposes. For instance, a number of previously developed optimal control methods based on geometric integrators [27, 36–39] could benefit from a formal method for establishing the resolution of discrete trajectories used for optimization.

Further work is necessary to provide guidelines for the practical application of the proposed bounds. While the derived upper limits for a single rigid body are simple and straightforward to use (i.e. they depend on the inertia condition number and norm of velocity), the situation with general multibody systems is more complex. We showed

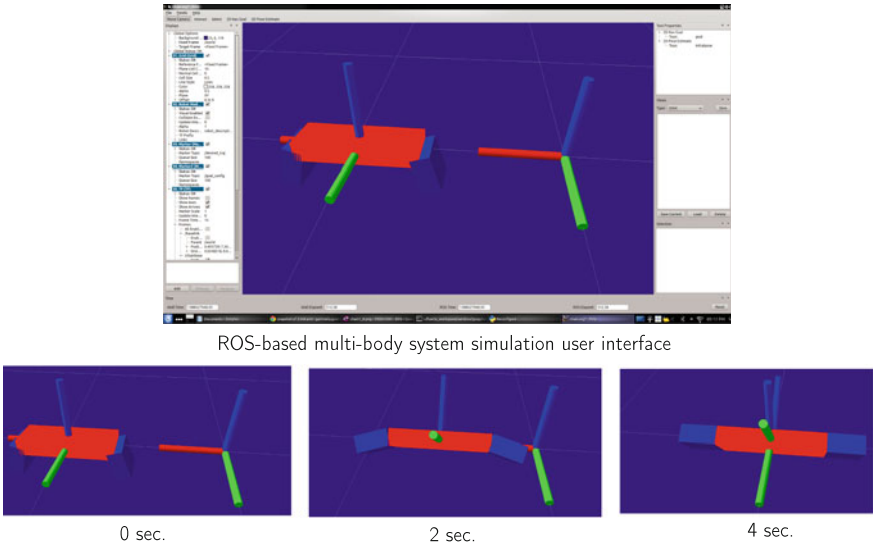


Fig. 7.4 A simple three-link chain with hinge joints simulated by the symplectic integrator and visualized using the robot operating system (ROS) user interface

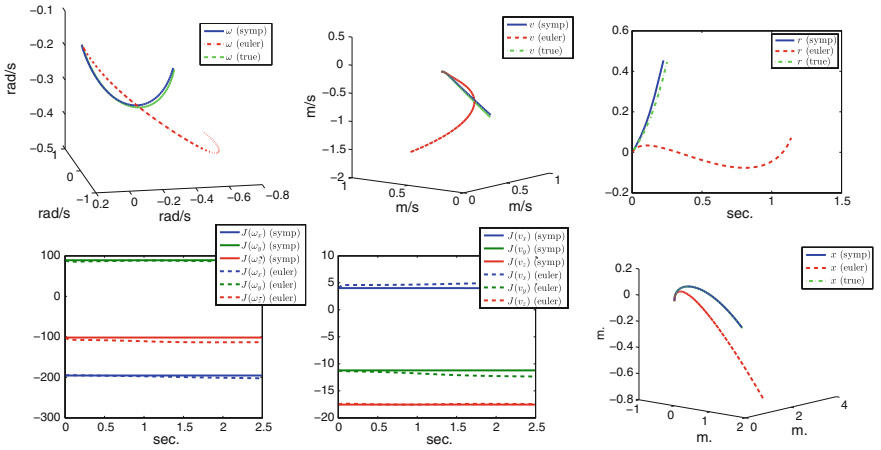


Fig. 7.5 The proposed symplectic integrators preserve the momentum map J and remain highly accurate even at large time-step (in this example $h = 0.1\,\mathrm{s}$). The key issue which remains to be answered is how high can h be chosen while retaining the solvability of the implicit time-stepping

that second-order geometric integrators in either generalized coordinates and or using $SE(3)$ matrices directly can be used as a basis for provably solvable time-stepping. Further study is necessary to establish a procedure for computing the dynamical model functions $\ell_0, \ldots, \ell_6$ based on the type of system under consideration.

While we considered variational integrators the proposed methodology could be extended to other energy-consistent low-order methods such as the discrete null space method [40]. A more challenging but equally important direction is to establish time-stepping bounds for more general systems involving intermittent contacts [41–47].

References

1. Marsden J, West M (2001) Discrete mechanics and variational integrators. Acta Numer 10:357–514
2. Kobilarov M, Crane K, Desbrun M (2009) Lie group integrators for animation and control of vehicles. ACM Trans Graph 28(2):1–14
3. Kharevych L, Weiwei Y, Tong E, Kanso J, Marsden P, Schroder, Desbrun M (2006) Geometric, variational integrators for computer animation. In: Eurographics/ACM SIGGRAPH symposium on computer, animation, pp 1–9
4. Barth E, Leimkuhler B (1993) Symplectic methods for conservative multibody systems. Fields Inst Commun 10:25–43
5. Reich S (1996) Symplectic integrators for systems of rigid bodies. In: Integration algorithms and classical mechanics, vol 10, p 383. Fields Institute Communications, AMS.
6. Jay L (1996) Symplectic partitioned rungekutta methods for constrained hamiltonian systems. SIAM J Numer Anal 33(1):368–387. [Online] Available http://epubs.siam.org/doi/abs/10.1137/0733019
7. Leyendecker S, Marsden JE, Ortiz M (2008) Variational integrators for constrained dynamical systems. Z Angew Math Mech 88(9):677–708
8. Johnson E, Murphey T (2009) Scalable variational integrators for constrained mechanical systems in generalized coordinates. IEEE Trans Robot 25(6):1249–1261
9. Kobilarov M, Marsden JE, Sukhatme GS (2010) Geometric discretization of nonholonomic systems with symmetries. AIMS J Discrete Continuous Dyn Syst Ser S (DCDS-S) 3(1):61–84
10. Betsch P, Hesch C, Sänger N, Uhlar S (2010) Variational integrators and energy-momentum schemes for flexible multibody dynamics. J Comput Nonlinear Dyn 5(3):031001
11. Barth E, Leimkuhler B, Reich S (1997) A semi-explicit, variable-stepsize, time-reversible integrator for constrained dynamics. SIAM J Sci Comput
12. Modin K, Fritzson D, Fuhrer C, Soderlind G (2005) A new class of variable step-size methods for multibody dynamics. In: ECCOMAS thematic conference on multibody dynamics 2005
13. Modin K, Fü hrer C (2006) Time-step adaptivity in variational integrators with application to contact problems. ZAMM 86(10):785–794
14. Holsapple R, Iyer R, Doman D (2007) Variable step-size selection methods for implicit integration schemes for odes. Int J Numer Anal Mod 4:210–240
15. Hairer E, Lubich C, Wanner G (2006) Geometric numerical integration. Springer series in computational mathematics, no 31. Springer, Berlin
16. Park F, Bobrow J, Ploen S (1995) A lie group formulation of robot dynamics. Int J Robot Res 14(6):609–618. [Online] Available http://ijr.sagepub.com/content/14/6/609.abstract
17. Mueller A, Maisser P (2003) A lie-group formulation of kinematics and dynamics of constrained mbs and its application to analytical mechanics. Multibody Syst Dyn 9:311–352
18. Park J, Chung W-K (2005) Geometric integration on euclidean group with application to articulated multibody systems. IEEE Trans Robot 21(5):850–863
19. Muller A, Terze Z (2009) Differential-geometric modelling and dynamic simulation of multibody systems. J Theor Appl Mech Eng 51(6)
20. Lanczos C (1949) Variational principles of mechanics. University of Toronto Press, Toronto
21. Marsden JE, Ratiu TS (1999) Introduction to mechanics and symmetry. Springer, Berlin
22. Krysl P, Endres L (2005) Explicit newmark/verlet algorithm for time integration of the rotational dynamics of rigid bodies. Int J Numer Methods Eng 62(15):2154–2177

23. Bou-Rabee N, Marsden J (2009) Hamilton-pontryagin integrators on Lie groups. Found Comput Math 9:197–219
24. Celledoni E, Owren B (2003) Lie group methods for rigid body dynamics and time integration on manifolds. Comput Meth Appl Mech Eng 19(3,4):421–438
25. Marsden JE, Pekarsky S, Shkoller S (1999) Discrete euler-poincare and Lie-poisson equations. Nonlinearity 12:16471662
26. Leok M (2004) Foundations of computational geometric mechanics. PhD dissertation, California Institute of Technology
27. Kobilarov M, Marsden J (2011) Discrete geometric optimal control on Lie groups. IEEE Trans Robot 27(4):641–655
28. Iserles A, Munthe-Kaas HZ, Nørsett SP, Zanna A (2000) Lie group methods. Acta Numer 9:215–365
29. Lewis F, Dawson D, Abdallah C (2003) Robot manipulator control: theory and practice, ser. Automation and Control Engineering. Taylor & Francis, [Online] Available http://books.google.com/books?id=8002tURlPP4C
30. Featherstone R (2008) Rigid body dynamics algorithms. Springer, Berlin
31. Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC, Boca Raton
32. Deuflhard P (2004) Newton methods for nonlinear problems: affine invariance and adaptive algorithms, ser. Springer series in computational mathematics. Springer, Berlin, Heidelberg, New York, autre tirage: 2006 [Online] Available http://opac.inria.fr/record=b1101121
33. Gragg WB, Tapia RA (1974) Optimal error bounds for the newton-kantorovich theorem. SIAM J Numer Anal 11(1):10–13. [Online] Available http://www.jstor.org/stable/2156425
34. Jain A (2011) Robot and multibody dynamics: analysis and algorithms. Springer, Berlin
35. Bobenko AI, Suris YB (1999) Discrete lagrangian reduction, discrete euler-poincare equations, and semidirect products. Lett Math Phys 49:79
36. de Leon M, de Diego DM, Santamaria Merino A (2004) Geometric numerical integration of nonholonomic systems and optimal control problems. Eur J Control 10:520–526
37. Lee T, Leok M, McClamroch N (2007) Lie group variational integrators for the full body problem in orbital mechanics. Celest Mech Dyn Astron 98(2):121–144
38. Bloch AM, Hussein II, Leok M, Sanyal AK (2009) Geometric structure-preserving optimal control of a rigid body. J Dyn Control Syst 15(3):307–330
39. Leyendecker S, Ober-Blbaum S, Marsden JE, Ortiz M (2010) Discrete mechanics and optimal control for constrained systems. Optimal Control Appl Methods 31(6):505–528. [Online] Available http://dx.doi.org/10.1002/oca.912
40. Betsch P, Leyendecker S (2006) The discrete null space method for the energy consistent integration of constrained mechanical systems. part ii: multibody dynamics. Int J Numeri Methods Eng 67(4):499–552. [Online]. Available http://dx.doi.org/10.1002/nme.1639
41. Pfeiffer F, Glocker C (1996) Multibody dynamics with unilateral contacts. Wiley series in nonlinear science
42. Anitescu M (2006) Optimization-based simulation of nonsmooth rigid multibody dynamics. Math Program 105:113–143. [Online] Available http://dx.doi.org/10.1007/s10107-005-0590-7
43. Potra FA, Anitescu M, Gavrea B, Trinkle J, A linearly implicit trapezoidal method for integrating stiff multibody dynamics with contact, joints, and friction. Int J Numer Methods Eng 66(7):1079–1124. [Online] Available http://dx.doi.org/10.1002/nme.1582
44. Studer C (2009) Numerics of unilateral contacts and friction, modeling and numerical time integration in non-smooth dynamics. Springer, Berlin
45. Koch MW, Leyendecker S (2011) Optimal control of multibody dynamics with contact. PAMM 11(1):51–52. [Online] Available http://dx.doi.org/10.1002/pamm.201110017
46. Jain A, Crean C, Kuo C, Bremen HV, Myint S (2012) Minimal coordinate formulation of contact dynamics in operational space. In: Robotics: science and systems
47. Chakraborty N, Berard S, Akella S, Trinkle J (2013) A geometrically implicit time-stepping method for multibody systems with intermittent contact. Int J Robot Res 32, no. tbd, p. tbd

Chapter 8
Variational Lie Group Formulation of Geometrically Exact Beam Dynamics: Synchronous and Asynchronous Integration

Thomas Leitz, Sina Ober-Blöbaum and Sigrid Leyendecker

Abstract For the elastodynamic simulation of a geometrically exact beam [1], an asynchronous variational integrator (AVI) [2] is derived from a PDE viewpoint. Variational integrators are symplectic and conserve discrete momentum maps and since the presented integrator is derived in the Lie group setting ($SO(3)$ for the representation of rotational degrees of freedom), it intrinsically preserves the group structure without the need for constraints [3]. The discrete Euler-Lagrange equations are derived in a general manner and then applied to the beam. A decrease of computational cost is to be expected in situations, where the time steps have to be very low in certain parts of the beam but not everywhere, e.g. if some regions of the beam are moving faster than others. The implementation allows synchronous as well as asynchronous time stepping and shows very good energy behavior, i.e. there is no drift of the total energy for conservative systems.

8.1 Introduction

The main subjects covered in this chapter are discrete mechanics and variational integrators, Lie group methods and beam dynamics. Their most recent developments are discussed in the following.

T. Leitz · S. Leyendecker (✉)
Applied Dynamics, University of Erlangen-Nuremberg, Haberstrasse 1,
91058 Erlangen, Germany
e-mail: sigrid.leyendecker@ltd.uni-erlangen.de

T. Leitz
e-mail: thomas.leitz@ltd.uni-erlangen.de

S. Ober-Blöbaum
Computational Dynamics and Optimal Control, Department of Mathematics,
University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
e-mail: sinaob@math.upb.de

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_8,

Discrete mechanics and variational integrators During the last years, the use of discrete mechanics for the construction of numerical discretization schemes has been of high interest to improve the simulation and optimization of mechanical systems. Typically, the underlying geometric structure of a mechanical system affects the qualitative behavior of solutions, and thus, numerical methods that preserve the geometry of a problem yield simulations that are qualitatively more accurate. Variational integrators are derived analogously to the equations of motion in continuous Lagrangian mechanics. To this end, principles and concepts from continuous mechanics are transferred to the discrete setting in such a way that the geometry is preserved in the discrete system which provides a geometric numerical integration scheme [4]. Instead of applying the continuous variational principle and discretizing the equations of motion, a discrete variational principle is used, which directly leads to the discrete Euler-Lagrange equations to integrate forward in time. As a result of the discrete variational principle, the symplectic structure of the Euler-Lagrange equations is carried over to the discrete setting. In consequence, the integrators are structure preserving, symplectic and a discrete Noether theorem can be proven as well as a realistic energy behavior, i.e. no numerical dissipation for conservative systems, is achieved. Variational integrators and their structure preserving properties are developed and analyzed by Marsden et al. [5] and have been further developed for many different classes of systems involving classical conservative mechanical systems (for an overview see [6, 7]), forced and controlled systems [8, 9], constrained systems (holonomic [10, 11] and nonholonomic systems [12]), nonsmooth systems [13], stochastic systems [14], multiscale systems [15, 16] and electric systems [17]. Variational integrators for Lagrangian PDE systems have been developed in [18]. In recent works [19], a variational framework is introduced that is based on generating functions for the simulation of Lagrangian PDE systems. To improve computational efficiency, asynchronous variational integrators are developed in [2] which allow the use of varying time step sizes throughout different elements in space, e.g. for different beam elements for the simulation of flexible beams. To further improve the computational efficiency, parallel asynchronous variational integrators are developed in [20]. Convergence results for the case of linear elastodynamics are shown in [21]. Applications of asynchronous variational integrators include the simulation of non linear hyperelastic solids in [22], contact mechanics in [23] and simulations using assumed gradient elements in [24].

Lie group methods For mechanical systems defined on Lie groups, Lie group methods are developed with the aim to preserve the Lie group structure of the system. Rather than introducing constraints and projection methods, which may destroy the good long-time behavior of the method, the corresponding group action is used to update the group element during the simulation such that the group structure is preserved in a natural way. Standard Lie group methods are the Crouch-Grossman (CG) methods [25] and the Runge-Kutta-Munthe-Kaas (RKMK) methods [26]. For an overview and a detailed discussion on Lie group methods we refer to [27] and the references therein.

The development of Lie group methods in variational integration theory is a recent research topic and addressed by several authors. Discrete Euler-Lagrange equations for systems on Lie groups and the associated discrete Lagrangian reductions have been carried out in [28–30] and further developed in [31–34] and applied to many examples. These integrators are referred to as Lie group variational integrators. The essential idea behind such integrators is to discretize Hamilton's principle by only allowing group operations when varying the discrete action and updating group elements using the exponential map of the Lie algebra. Lie group formulations are in particular appropriate to describe the orientation of rigid bodies or cross sections of flexible beams which motivates recent works on variational Lie group integrators for the structure-preserving simulation of geometrically exact beams as e.g. in [35, 36].

Beam dynamics Modeling geometrically exact beams as a special Cosserat continuum (see e.g., [1]) has been the basis for many finite element formulations starting with [37]. The formulation of beam dynamics as a Lagrangian system requires a representation of the rotational degrees of freedom and their kinematics, which can, on the one hand, be treated by a local parametrization of the Lie group $SO(3)$ or, on the other hand, by using a redundant configuration variable which is subject to constraints. Many current semi-discrete beam formulations avoid the introduction of constraints by using rotational degrees of freedom, see, e.g., [38, 39]. However, it has been shown by Crisfield and Jelenić [40], that the interpolation of non commutative finite rotations may destroy the objectivity of the strain measures in the semi discrete model. To overcome this problem, a director triad can be introduced, which is constrained to be orthonormal in each node of the central line of the beam; thus it forms the columns of the rotational matrix. The spatial interpolation of the director triad leads to objective strain measures in the spatially discretized configuration (even though the interpolated directors might fail to be orthonormal). This idea is independently developed in [41] and [42]. For an overview on the effects of different interpolation techniques concerning frame invariance and the appearance of singularities we refer to [43]. Furthermore, this subject is elaborated in [44–48]. The constrained formulation is particularly popular when the beam is part of a multibody system, where further constraints representing the connection to other (rigid or flexible) components are naturally present. Another common formulation is the so called absolute nodal coordinates formulation based on works of [49, 50]. Recently, Lie group formulations are becoming more and more important in multibody and beam dynamics; see, e.g., [51, 52]. So far, only few works exist on beam dynamics simulation based on Lie group variational integrators (see e.g. [35, 36]). In this work, we derive a variational Lie group integrator, similar to the procedure described in [36]. However, here the focus is on a formulation that is directly applicable in practical implementation, thus particular care is given to the formulation of discrete conjugate momenta and boundary conditions. The discrete formulation of the Lagrangian system allows the construction of synchronous as well as asynchronous variational time stepping schemes.

Outline In Sect. 8.2, group structure preserving variations on $SO(3)$ are introduced using a one parameter sub group element and the directional derivative. These variations are then used to derive the continuous Euler-Lagrange equations for a Lagrangian of a mechanical system defined on the Lie group $SO(3)$. The conjugate momenta in space and time are given and interpreted for classical mechanics. In Sect. 8.3, the synchronous discrete Euler-Lagrange equations are derived for a discrete Lagrangian defined on a regular space time grid. The discrete conjugate momenta are given along with their connection to the continuous conjugate momenta. Special attention is paid to the correct boundary conditions in space and time. A method for solving the discrete Euler-Lagrange equations without the need for orthonormality constraints is presented. Quadrature rules for the evaluation of the discrete Lagrangian are introduced and later generalized to the asynchronous case. In Sect. 8.4, the Lagrangian density for the geometrically exact beam dynamics is discussed and the continuous Euler-Lagrange equations are derived. The boundary conditions are interpreted and discretizations of the time and space derivatives on $SO(3)$ and $\mathbb{R}^3$ are given that lead to the discrete Lagrangian for the derivation of the AVI. As an example, the results of the simulation of a beam are shown in Sect. 8.5, where the conservation of momentum maps following from the discrete Noether theorem is illustrated.

8.2 Hamilton's Principle on $SO(3)$

In this section, Hamilton's principle is formulated for a space and time dependent system on the Lie group $SO(3)$ to derive the Euler-Lagrange equations. In this case, the Euler-Lagrange equations are partial differential equations of motion for a dynamical system in which the spatially continuously distributed configurations depend on translational and rotational degrees of freedom. To this end, variations respecting the structure of the Lie group are defined in Sect. 8.2.1 before the variational principle itself is formulated in Sect. 8.2.2.

8.2.1 Group Structure Preserving Variations

When computing the variation of a function $f:SO(3) \to \mathbb{R}$ in terms of the variation of the group element, special care has to be taken in order not to violate the group structure. For example, treating orthonormal matrices Λ as elements of the vector space $\mathbb{R}^{3\times3}$ (where the natural operation is matrix addition) leads to a violation of the group structure of $SO(3)$. To illustrate this, consider the variation in the linear space

$$\delta f\left(\Lambda\right) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0} f\left(\Lambda + \varepsilon\delta\Lambda\right) = \frac{\partial f}{\partial \Lambda} : \delta\Lambda$$

using the small parameter $\varepsilon \in \mathbb{R}$. Performing the operation $\Lambda + \varepsilon\delta\Lambda$ is not compatible with the group structure of $SO(3)$ (where the natural operation is matrix

multiplication). This problem can, for instance, be compensated by applying orthonormality constraints or a projection method when solving the equations $\delta f = 0$. In contrast to that, the group structure is inherently respected by performing variations in terms of the group operation. Consequently, the variation of Λ is given by

$$\delta\Lambda = \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} \Lambda\delta\Lambda_\varepsilon$$

where $\delta\Lambda_\varepsilon$ can be expressed by the exponential map of a skew symmetric matrix $\varepsilon\hat{\eta}$ in the Lie algebra $\mathfrak{so}(3)$. The hat map $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ defined as $\hat{a}_{ik} = \epsilon_{ijk} a_j$ and its inverse $\left(\hat{a}\right)^\vee_j = \frac{1}{2}\epsilon_{ijk}\hat{a}_{ik}$, where ϵ_{ijk} is the Levi-Civita symbol, serve as the isomorphism $\mathfrak{so}(3) \simeq \mathbb{R}^3$. Using Rodrigues' formula and $\eta = \Theta n$, where $\Theta \in \mathbb{R}$ is the rotation angle and $n \in \mathbb{R}^3$ is a three dimensional unit vector, the variation of Λ reads

$$\begin{aligned}\delta\Lambda = \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} \left(\Lambda e^{\varepsilon\hat{\eta}}\right) &= \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} \left(\Lambda\left(I + \hat{n}\sin(\varepsilon\Theta) + (1-\cos(\varepsilon\Theta))\, n\otimes n\right)\right) \\ &= \Lambda\left(\hat{n}\Theta\cos(\varepsilon\Theta) + \Theta\sin(\varepsilon\Theta)\, n\otimes n\right)\Big|_{\varepsilon=0} \\ &= \Lambda\left(\Theta\hat{n}\right) = \Lambda\hat{\eta}.\end{aligned}$$

Using this, the variation of the rotation dependent function is given by

$$\begin{aligned}\delta f(\Lambda) = \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} f\left(\Lambda e^{\varepsilon\hat{\eta}}\right) &= \frac{\partial f}{\partial\Lambda} : \left(\Lambda\hat{\eta}\right) = \left(\Lambda^T\frac{\partial f}{\partial\Lambda}\right)^{(A)} : \hat{\eta} \\ &= 2\left[\left(\Lambda^T\frac{\partial f}{\partial\Lambda}\right)^{(A)}\right]^\vee \cdot \eta.\end{aligned}$$

Here, we use the fact, that double contraction[1] of a symmetric matrix with a skew symmetric matrix is zero, and thus only the skew symmetric part, denoted by the superscript (A), remains. Using the above, the variation of functions of the angular velocity $\hat{\omega} = \Lambda^T\dot{\Lambda}$ and the angular strain $\hat{\Omega} = \Lambda^T\Lambda'$ (to be introduced in Sect. 8.2.2) can be found as

$$\delta f\left(\hat{\omega}\right) = \left.\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\right|_{\varepsilon=0} f\left(\left(\Lambda e^{\varepsilon\hat{\eta}}\right)^T\frac{\mathrm{d}}{\mathrm{d}t}\left(\Lambda e^{\varepsilon\hat{\eta}}\right)\right) = \frac{\partial f}{\partial\hat{\omega}} : \left(-\hat{\eta}\hat{\omega} + \hat{\omega}\hat{\eta} + \dot{\hat{\eta}}\right)$$

and thus

$$\delta f(\omega) = \frac{\partial f}{\partial\omega}\cdot\left(\omega\times\eta + \dot{\eta}\right) \quad \delta f(\Omega) = \frac{\partial f}{\partial\Omega}\cdot\left(\Omega\times\eta + \eta'\right).$$

[1] The symbol ':' represents a scalar matrix-matrix product with the summation over two indices, i.e. $A{:}B = A_{ij}B_{ij} \in \mathbb{R}$ for two matrices of matching dimension.

8.2.2 Continuous Euler-Lagrange Equations

Consider a one dimensional reference configuration space variable $s \in [0, \ell] \subset \mathbb{R}$, the time variable $t \in [0, T] \subset \mathbb{R}$ and the deformation map $\varphi\colon (s, t) \mapsto (\Lambda, x) \in SO(3) \times \mathbb{R}^3$. The Lagrange density $L\left(\Lambda, \omega, \Omega, x, \dot{x}, x'\right)$ is a function of the configuration (Λ, x) being composed by orientation and position in the ambient space, the angular velocity $\hat{\omega} = \Lambda^T \dot{\Lambda} \in \mathfrak{so}(3)$ and bending and torsional strain $\hat{\Omega} = \Lambda^T \Lambda' \in \mathfrak{so}(3)$ as well as the translational velocity $\dot{x}$ and the shear and elongational strain x', where

$$\dot{(\cdot)} = \frac{\mathrm{d}(\cdot)}{\mathrm{d}t} \quad (\cdot)' = \frac{\mathrm{d}(\cdot)}{\mathrm{d}s}.$$

The action functional is defined as the integral of the Lagrange density over space and time

$$S[\varphi] = \int_0^T \int_0^\ell L \mathrm{d}s \mathrm{d}t.$$

For convenience, we split the variation of the action into two parts $\delta S = \delta_\Lambda S + \delta_x S$. Here, $\delta_\Lambda S$ represents the variation with respect to orientation and $\delta_x S$ represents the variation with respect to position. Using integration by parts in space and time and the variations of orientation $\delta\Lambda = \Lambda\hat{\eta}$, of the angular velocity $\delta\omega = \omega \times \eta + \dot{\eta}$ and of the angular strain $\delta\Omega = \Omega \times \eta + \eta'$, respectively, yields the variations of the action

$$\begin{aligned}
\delta_\Lambda S = &\int_0^T \int_0^\ell \left[2\left[\left(\Lambda^T \frac{\partial L}{\partial \Lambda}\right)^{(A)} \right]^\vee - \omega \times \frac{\partial L}{\partial \omega} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \omega}\right) \right. \\
&\left. - \Omega \times \frac{\partial L}{\partial \Omega} - \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial L}{\partial \Omega}\right) \right] \cdot \eta \mathrm{d}s \mathrm{d}t \\
&+ \int_0^\ell \frac{\partial L}{\partial \hat{\omega}} \cdot \hat{\eta} \mathrm{d}s \Big|_0^T + \int_0^T \frac{\partial L}{\partial \hat{\Omega}} \cdot \hat{\eta} \mathrm{d}t \Big|_0^\ell \\
\delta_x S = &\int_0^T \int_0^\ell \left[\frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial L}{\partial x'}\right) \right] \cdot \delta x \mathrm{d}s \mathrm{d}t \\
&+ \int_0^\ell \frac{\partial L}{\partial \dot{x}} \cdot \delta x \mathrm{d}s \Big|_0^T + \int_0^T \frac{\partial L}{\partial x'} \cdot \delta x \mathrm{d}t \Big|_0^\ell .
\end{aligned}$$

Following Hamilton's principle, the action is stationary with respect to all variations while holding the boundaries of space and time fixed, i.e. $\delta\Lambda = 0$, $\delta x = 0$ for $t = 0$, $t = T$, $s = 0$ and $s = \ell$. The resulting Euler-Lagrange equations are

$$\begin{bmatrix} 2\left[\left(\Lambda^T \frac{\partial L}{\partial \Lambda}\right)^{(A)}\right]^{\vee} - \omega \times \frac{\partial L}{\partial \omega} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \omega}\right) - \Omega \times \frac{\partial L}{\partial \Omega} - \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial L}{\partial \Omega}\right) \\ \frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial L}{\partial x'}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.1}$$

This coupled system of partial differential equations represents a local balance of angular and linear momentum and needs to be solved with appropriate boundary conditions as will be discussed in later sections. The temporal and spatial Legendre transforms for the rotational and translational part represent the conjugate momenta of time and space

$$\Pi = \frac{\partial L}{\partial \omega} \quad \Gamma = \frac{\partial L}{\partial \dot{x}} \quad \Sigma = \frac{\partial L}{\partial \Omega} \quad \sigma = \frac{\partial L}{\partial x'}. \tag{8.2}$$

In classical mechanics, Π and Γ are angular and linear momenta per unit length while Σ represents bending and torsional torques and σ are normal and shear forces, see also Sect. 8.4 on geometrically exact beam dynamics.

8.3 Discrete Hamilton's Principle on $SO(3)$

The key feature of variational integrators is that they are based on a discrete variational formulation of the underlying system, e.g. a discrete version of Hamilton's principle for conservative mechanical systems, instead of a discretization of the equations of motion. The variational theory of discrete mechanics provides a theoretical framework that parallels continuous variational dynamics. Discrete analogues to the Euler-Lagrange equations, the symplectic structure and Noether's theorem are derived from a discrete Lagrangian by performing similar steps as in the continuous theory. A detailed introduction and a survey on the history and literature on the variational view of discrete mechanics is given in [5]. Before deriving discrete analogues of the partial differential Euler-Lagrange equations (8.1), the concept of variational integrators is briefly illustrated for a finite dimensional dynamical system.

Discrete variational principle in finite dimensions Let the Lagrangian $L(q, \dot{q})$ be defined in terms of the configuration variable $q \in \mathbb{R}^n$ and its velocity. Let $t^0, t^0 + \Delta t, \ldots, t^0 + N\Delta t = t_N$ denote an equidistant time grid with time step $\Delta t \in \mathbb{R}$. In a time interval $[t^j, t^{j+1}]$, the action integral is approximated using the

discrete Lagrangian $L^j \approx \int_{t^j}^{t^{j+1}} L(q, \dot{q})\mathrm{d}t$. The quadrature used to approximate the integral determines the actual time stepping scheme (8.3) and in particular its order of accuracy. The action integral is approximated by the discrete action sum $S_d = \sum_{j=0}^{N-1} L^j$. Requiring its stationarity for all variations vanishing on the boundary, i.e. $\delta q^0 = \delta q^N = 0$, yields the discrete Euler-Lagrange equations for $j = 1, \ldots, N-1$

$$\frac{\partial S_d}{\partial q^j} = \frac{\partial L^j}{\partial q^j} + \frac{\partial L^{j-1}}{\partial q^j} = 0. \tag{8.3}$$

For a given initial configuration $q^0 = q(t^0)$ and initial velocity $\dot{q}(t^0)$ with corresponding initial conjugate momentum derived via the Legendre transform

$$p^0 = p(t^0) = \frac{\partial L(q(t^0), \dot{q}(t^0))}{\partial \dot{q}}$$

the first discrete configuration q^1 can be computed by solving $p^0 = -\frac{\partial L^0}{\partial q^0}$. Then, for two given subsequent configurations, (8.3) can be used to integrate forward in time. More generally, defining the discrete conjugate momenta in time for $j = 0, \ldots, N$ via the discrete Legendre transforms on the future time interval $\left(p^j\right)^- = -\frac{\partial L^j}{\partial q^j}$ and on the past time interval $\left(p^j\right)^+ = \frac{\partial L^{j-1}}{\partial q^j}$, the time stepping scheme (8.3) can be interpreted as a matching of the discrete conjugate momenta $\left(p^j\right)^- = \left(p^j\right)^+$. See Refs. [5, 6] for the theory on (discrete) Legendre transforms. The resulting integrators are symplectic and momentum-preserving and have an excellent long-time energy behavior for constant time steps, which can be shown by means of backward error analysis [4].

8.3.1 Discrete Euler-Lagrange Equations

In order to derive a synchronous variational integrator, space time is discretized on a regular grid as shown in Fig. 8.1. This grid is not necessarily equidistant in space, though the time step is kept constant in order to allow backward error analysis [4]. Thus, all spatial elements advance in time synchronously. The discrete Lagrangian is an approximation of the continuous action for one space time element K_a^j, i.e.

$$L_a^j \approx \int_{t^j}^{t^{j+1}} \int_{s_a}^{s_{a+1}} L\mathrm{d}s\mathrm{d}t \tag{8.4}$$

and it is generally a function of all four nodes of the space time element K_a^j

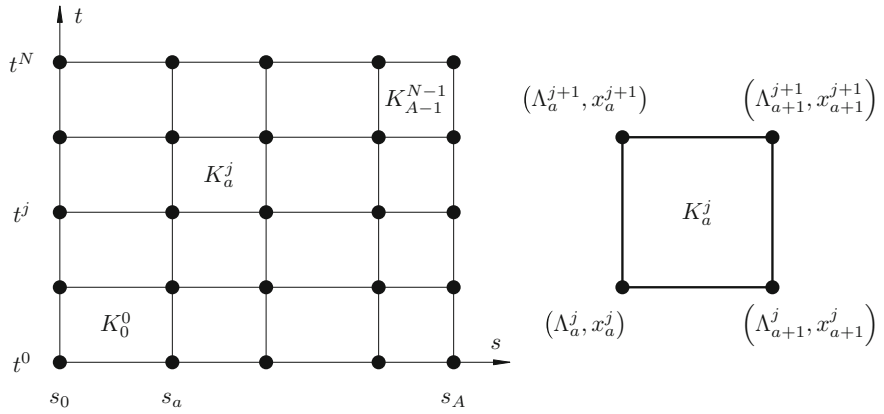


Fig. 8.1 Regular discretization of the space time for synchronous variational integrators

$$L_a^j = L_a^j\left(\Lambda_a^j, \Lambda_a^{j+1}, \Lambda_{a+1}^j, \Lambda_{a+1}^{j+1}, x_a^j, x_a^{j+1}, x_{a+1}^j, x_{a+1}^{j+1}\right). \tag{8.5}$$

Thus, the discrete action sum is the sum over all discrete Lagrangians

$$S_d = \sum_{j=0}^{N-1}\sum_{a=0}^{A-1} L_a^j. \tag{8.6}$$

In general, the discrete action is a function of all nodes $\left(\Lambda_a^j, x_a^j\right)$ in space time, thus the discrete variation leads to

$$\begin{aligned}\delta S_d &= \sum_{a=0}^{A}\sum_{j=0}^{N}\left(\frac{\partial S_d}{\partial \Lambda_a^j} : \delta\Lambda_a^j + \frac{\partial S_d}{\partial x_a^j}\cdot\delta x_a^j\right)\\ &= \sum_{a=0}^{A}\sum_{j=0}^{N}\left(2\left[\left(\left(\Lambda_a^j\right)^T\frac{\partial S_d}{\partial \Lambda_a^j}\right)^{(A)}\right]^{\vee}\cdot\eta_a^j + \frac{\partial S_d}{\partial x_a^j}\cdot\delta x_a^j\right)\end{aligned}$$

and the discrete Hamilton's principle states, that the discrete action is stationary for all variations vanishing on the boundary of space and time, i.e. $\eta_a^j = 0$ and $\delta x_a^j = 0$ for $a = 0$, $a = A$, $j = 0$, $j = N$. Thus, discrete Euler-Lagrange equations for $a = 1, \ldots, A-1$ and $j = 1, \ldots, N-1$ are

$$2\left[\left(\left(\Lambda_a^j\right)^T\frac{\partial S_d}{\partial \Lambda_a^j}\right)^{(A)}\right]^{\vee} = 0 \quad \frac{\partial S_d}{\partial x_a^j} = 0. \tag{8.7}$$

Inserting the specific discrete action composed of the discrete Lagrangians (8.6) leads to the discrete version of (8.1)

$$\begin{bmatrix} 2\left[\left(\left(\Lambda_a^j\right)^T \frac{\partial L_a^j}{\partial \Lambda_a^j} + \left(\Lambda_a^j\right)^T \frac{\partial L_a^{j-1}}{\partial \Lambda_a^j} + \left(\Lambda_a^j\right)^T \frac{\partial L_{a-1}^j}{\partial \Lambda_a^j} + \left(\Lambda_a^j\right)^T \frac{\partial L_{a-1}^{j-1}}{\partial \Lambda_a^j}\right)^{(A)}\right]^\vee \\ \frac{\partial L_a^j}{\partial x_a^j} + \frac{\partial L_a^{j-1}}{\partial x_a^j} + \frac{\partial L_{a-1}^j}{\partial x_a^j} + \frac{\partial L_{a-1}^{j-1}}{\partial x_a^j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.8}$$

which has to be supplemented by boundary conditions. Let Π, Γ, Σ and σ denote conjugate momenta in the time and space continuous setting and define

$$\Pi^j = \Pi\left(s, t^j\right) \quad \Gamma^j = \Gamma\left(s, t^j\right)$$

for $s \in [0, \ell]$ and

$$\Sigma_a = \Sigma\left(s_a, t\right) \quad \sigma_a = \sigma\left(s_a, t\right)$$

for $t \in [0, T]$. Using

$$F_\Lambda \bar{L} = 2\left[\left(\left(\Lambda_a^j\right)^T \frac{\partial}{\partial \Lambda_a^j}\left(\bar{L}\right)\right)^{(A)}\right]^\vee \quad F_x \bar{L} = \frac{\partial}{\partial x_a^j}\left(\bar{L}\right)$$

for a discrete Lagrangian $\bar{L}$ of the form (8.5), the discrete Legendre transforms with respect to orientation Λ_a^j and position x_a^j are defined as $F\bar{L} = \left[F_\Lambda \bar{L}, F_x \bar{L}\right]^T$. Note that there exist four index combinations, for which $\bar{L}$ depends on (Λ_a^j, x_a^j).The discrete Legendre transforms are approximations of the following integrals

$$FL_{a-1}^j \approx -\int_{s_{a-\frac{1}{2}}}^{s_a} \begin{bmatrix} \Pi^j \\ \Gamma^j \end{bmatrix} \mathrm{d}s + \int_{t^j}^{t^{j+\frac{1}{2}}} \begin{bmatrix} \Sigma_a \\ \sigma_a \end{bmatrix} \mathrm{d}t \quad FL_a^j \approx -\int_{s_a}^{s_{a+\frac{1}{2}}} \begin{bmatrix} \Pi^j \\ \Gamma^j \end{bmatrix} \mathrm{d}s - \int_{t^j}^{t^{j+\frac{1}{2}}} \begin{bmatrix} \Sigma_a \\ \sigma_a \end{bmatrix} \mathrm{d}t$$

$$FL_{a-1}^{j-1} \approx \int_{s_{a-\frac{1}{2}}}^{s_a} \begin{bmatrix} \Pi^j \\ \Gamma^j \end{bmatrix} \mathrm{d}s + \int_{t^{j-\frac{1}{2}}}^{t^j} \begin{bmatrix} \Sigma_a \\ \sigma_a \end{bmatrix} \mathrm{d}t \quad FL_a^{j-1} \approx \int_{s_a}^{s_{a+\frac{1}{2}}} \begin{bmatrix} \Pi^j \\ \Gamma^j \end{bmatrix} \mathrm{d}s - \int_{t^{j-\frac{1}{2}}}^{t^j} \begin{bmatrix} \Sigma_a \\ \sigma_a \end{bmatrix} \mathrm{d}t.$$

For $j = 0, \ldots, N$ and $a, = 0 \ldots, A$ the discrete conjugate momenta in time are given by

$$\begin{bmatrix} \left(\Pi_a^j\right)^- \\ \left(\Gamma_a^j\right)^- \end{bmatrix} = -\left(FL_{a-1}^j + FL_a^j\right) \quad \begin{bmatrix} \left(\Pi_a^j\right)^+ \\ \left(\Gamma_a^j\right)^+ \end{bmatrix} = FL_{a-1}^{j-1} + FL_a^{j-1} \tag{8.9}$$

while discrete conjugate momenta in space read

$$\begin{bmatrix} \left(\Sigma_a^j\right)^- \\ \left(\sigma_a^j\right)^- \end{bmatrix} = -\left(FL_a^j + FL_a^{j-1}\right) \quad \begin{bmatrix} \left(\Sigma_a^j\right)^+ \\ \left(\sigma_a^j\right)^+ \end{bmatrix} = FL_{a-1}^j + FL_{a-1}^{j-1}. \tag{8.10}$$

The discrete Euler-Lagrange equations (8.8) can now equivalently be written as

$$\begin{bmatrix} \left(\Pi_a^j\right)^- \\ \left(\Gamma_a^j\right)^- \end{bmatrix} = \begin{bmatrix} \left(\Pi_a^j\right)^+ \\ \left(\Gamma_a^j\right)^+ \end{bmatrix} \tag{8.11}$$

or equivalently

$$\begin{bmatrix} \left(\Sigma_a^j\right)^- \\ \left(\sigma_a^j\right)^- \end{bmatrix} = \begin{bmatrix} \left(\Sigma_a^j\right)^+ \\ \left(\sigma_a^j\right)^+ \end{bmatrix}.$$

Whenever $j = 0$, $j = N$, $a = 0$ or $a = A$ in (8.9) and (8.10), certain boundary conditions have to be considered. On the boundary of space, these are the time integrals over boundary values of the spatial conjugate momenta

$$\begin{aligned} FL_{-1}^j &= \int_{t^j}^{t^{j+\frac{1}{2}}} \begin{bmatrix} \Sigma_0 \\ \sigma_0 \end{bmatrix} \mathrm{d}t \quad FL_A^j = -\int_{t^j}^{t^{j+\frac{1}{2}}} \begin{bmatrix} \Sigma_A \\ \sigma_A \end{bmatrix} \mathrm{d}t \\ FL_{-1}^{j-1} &= \int_{t^{j-\frac{1}{2}}}^{t^j} \begin{bmatrix} \Sigma_0 \\ \sigma_0 \end{bmatrix} \mathrm{d}t \quad FL_A^{j-1} = -\int_{t^{j-\frac{1}{2}}}^{t^j} \begin{bmatrix} \Sigma_A \\ \sigma_A \end{bmatrix} \mathrm{d}t. \end{aligned} \tag{8.12}$$

The temporal boundary conditions are the space integrals over the temporal conjugate momenta

$$\begin{aligned} FL_{a-1}^N &= -\int_{s_{a-\frac{1}{2}}}^{s_a} \begin{bmatrix} \Pi^N \\ \Gamma^N \end{bmatrix} \mathrm{d}s \quad FL_a^N = -\int_{s_a}^{s_{a+\frac{1}{2}}} \begin{bmatrix} \Pi^N \\ \Gamma^N \end{bmatrix} \mathrm{d}s \\ FL_{a-1}^{-1} &= \int_{s_{a-\frac{1}{2}}}^{s_a} \begin{bmatrix} \Pi^0 \\ \Gamma^0 \end{bmatrix} \mathrm{d}s \quad FL_a^{-1} = \int_{s_a}^{s_{a+\frac{1}{2}}} \begin{bmatrix} \Pi^0 \\ \Gamma^0 \end{bmatrix} \mathrm{d}s \end{aligned} \tag{8.13}$$

as illustrated in Fig. 8.2, where the notation q_a^j comprises Λ_a^j and x_a^j. The Legendre transforms on the corners of space time are always zero, i.e.

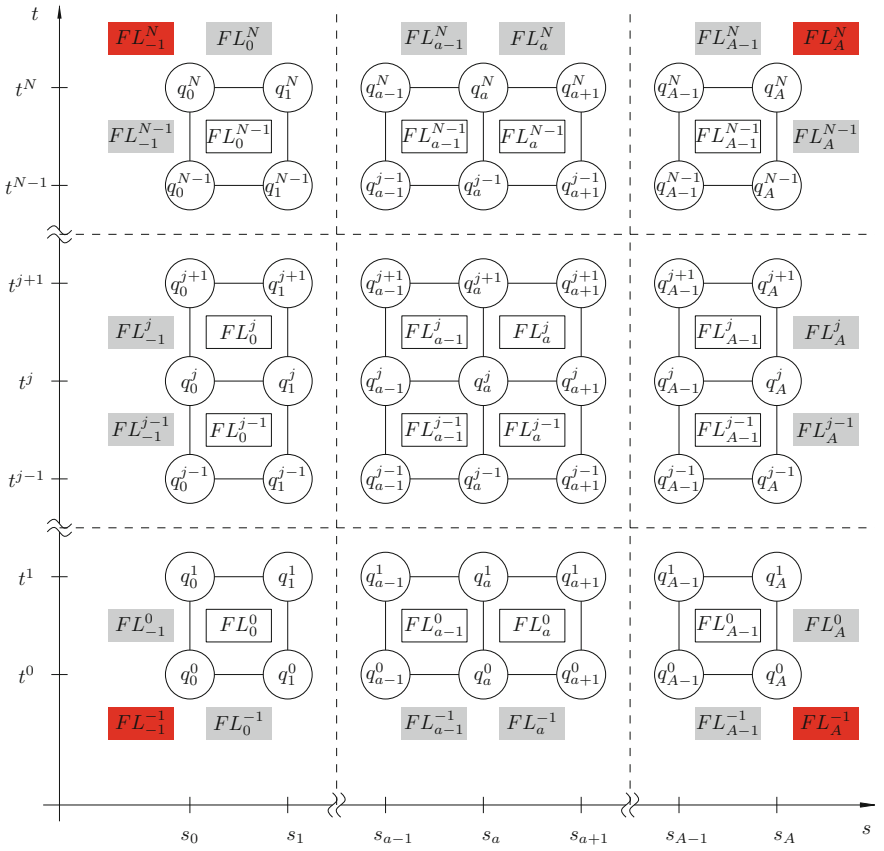


Fig. 8.2 Visualization of the general Legendre transforms and boundary conditions. The boundary conditions are *gray* and on the corners of space time they are zero (*red*). The notation q_a^j comprises Λ_a^j and x_a^j

$$FL_{-1}^{-1} = FL_{-1}^{N} = FL_{A}^{-1} = FL_{A}^{N} = 0.$$

The Legendre transform FL_a^j is generally a function of the four nodes in the discrete Lagrangian from which it is derived, i.e. $FL_a^j = FL_a^j\left(q_a^j, q_a^{j+1}, q_{a+1}^j, q_{a+1}^{j+1}\right)$. Since each node has six degrees of freedom and there are $A+1$ spatial nodes in the grid, to advance one step in time, the integrator (8.11) is a system of coupled non linear equations for all spatial nodes, i.e. it is of the size $6(A+1)$ and has to be solved at once in each time step. However, certain approximations of time and space derivatives and quadrature rules for the discrete Lagrangian (see Sect. 8.3.3), for which the discrete Legendre transforms with respect to q_a^j are independent of the corner nodes of the rectangle surrounding q_a^j, i.e. for which

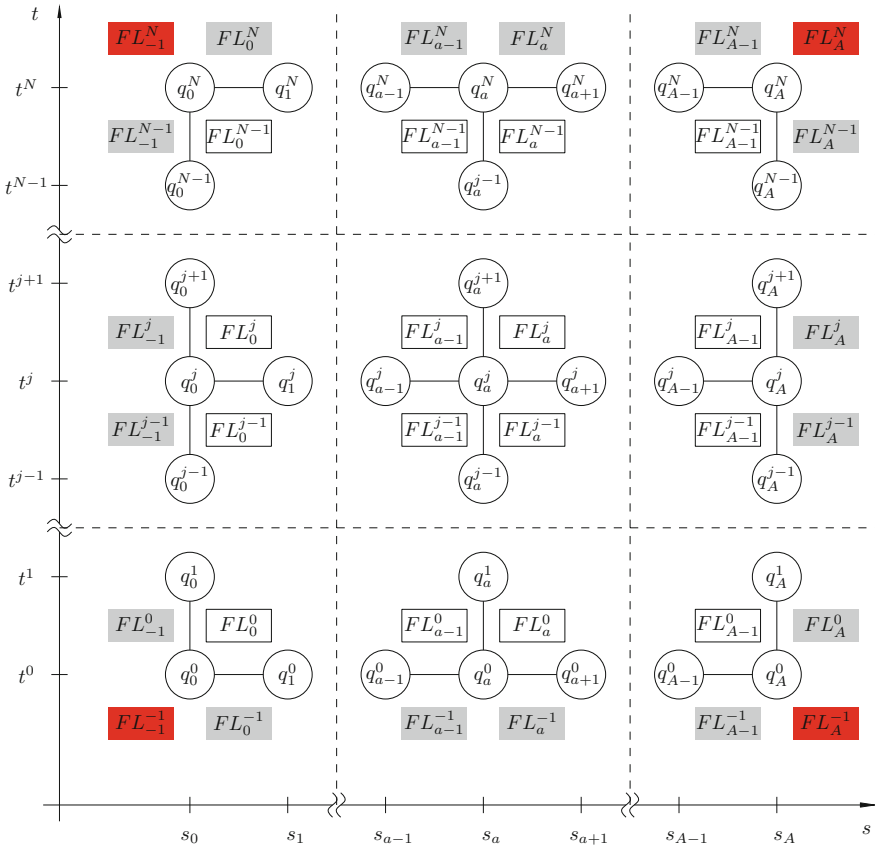


Fig. 8.3 Visualization of the special Legendre transforms, where (8.14) holds true, and boundary conditions. The boundary conditions are *gray* and on the corners of space time they are zero (*red*). The notation q_a^j comprises Λ_a^j and x_a^j

$$\frac{\partial FL_{a-1}^j}{\partial q_{a-1}^{j+1}} = \frac{\partial FL_a^j}{\partial q_{a+1}^{j+1}} = \frac{\partial FL_{a-1}^{j-1}}{\partial q_{a-1}^{j-1}} = \frac{\partial FL_a^{j-1}}{\partial q_{a+1}^{j-1}} = 0 \tag{8.14}$$

holds true,[2] lead to $A+1$ independent non linear systems of equations for each node of size six that can be solved individually as shown in Fig. 8.3.

[2] Note that $\frac{\partial(\cdot)}{\partial q}$ denotes $\frac{\partial(\cdot)}{\partial x}$ and $2\left[\left(\Lambda^T \frac{\partial(\cdot)}{\partial \Lambda}\right)^{(A)}\right]^\vee$, respectively.

8.3.2 Solving the Discrete Euler-Lagrange Equations

The discrete Euler-Lagrange equations for one time step (8.8), or equaivalently (8.11), are the system of non linear equations

$$R^{\Lambda}\left(\Lambda_0^{j+1}, x_0^{j+1}, \ldots, \Lambda_A^{j+1}, x_A^{j+1}\right) = \begin{bmatrix} R_0^{\Lambda} \\ \vdots \\ R_A^{\Lambda} \end{bmatrix} = 0 \in \mathbb{R}^{6(A+1)} \tag{8.15}$$

with the nodal discrete Euler-Lagrange equations

$$\begin{aligned} &R_0^{\Lambda}\left(\Lambda_0^{j+1}, x_0^{j+1}, \Lambda_1^{j+1}, x_1^{j+1}\right) = 0 \\ &R_a^{\Lambda}\left(\Lambda_{a-1}^{j+1}, x_{a-1}^{j+1}, \Lambda_a^{j+1}, x_a^{j+1}, \Lambda_{a+1}^{j+1}, x_{a+1}^{j+1}\right) = 0 \quad \text{for } 1 \le a \le A-1 \\ &R_A^{\Lambda}\left(\Lambda_{A-1}^{j+1}, x_{A-1}^{j+1}, \Lambda_A^{j+1}, x_A^{j+1}\right) = 0. \end{aligned}$$

In case that (8.14) holds true, the system decouples to independent sets of six equations for each node $a = 0, \ldots, A$

$$R_a^{\Lambda}\left(\Lambda_a^{j+1}, x_a^{j+1}\right) = 0 \in \mathbb{R}^6.$$

In any case, there are twelve unknowns and only six discrete Euler-Lagrange equations for each node in the grid, if one regards the orientation naively as a 3×3 matrix. In order to avoid orthonormality constraints to ensure that $\Lambda_a^{j+1} \in SO(3)$, we use the reparametrization $\Lambda_a^{j+1} = \Lambda_a^j \operatorname{Cay}(\hat{f}_a^j)$ where Cay is the Cayley map and $\hat{f}_a^j \in \mathfrak{so}(3)$ is the increment. The singularity of the Cayley map is avoided by the fact that the rotation increment $\operatorname{Cay}(f_a^j)$ is close to the identity for small time steps and thus far away from rotation about the angle π. The nodal discrete Euler-Lagrange equations are transformed to

$$R_a^f\left(f_a^j, x_a^{j+1}\right) = 0 \in \mathbb{R}^6$$

and the same transformation is possible for the fully coupled system (8.15). The equations can be solved for f_a^j and x_a^{j+1} by a Newton-Raphson scheme and the new orientation is recovered by $\Lambda_a^{j+1} = \Lambda_a^j \operatorname{Cay}(\hat{f}_a^j)$. Note that the evaluation of the Cayley map does not involve any transcendental functions. In terms of the components f_1, f_2, f_3 of f, it can explicitly be written as

$$\operatorname{Cay}\left(\hat{f}\right) = \left(I - \hat{f}\right)^{-1}\left(I + \hat{f}\right)$$

$$= \frac{1}{f_1{}^2 + f_2{}^2 + f_3{}^2 + 1} \begin{pmatrix} f_1{}^2 - f_2{}^2 - f_3{}^2 + 1 & -2(f_3 - f_1 f_2) & 2(f_2 + f_1 f_3) \\ 2(f_3 + f_1 f_2) & -f_1{}^2 - f_2{}^2 + f_3{}^2 - 1 & -2(f_1 - f_2 f_3) \\ -2(f_2 - f_1 f_3) & 2(f_1 + f_2 f_3) & -f_1{}^2 + f_2{}^2 - f_3{}^2 - 1 \end{pmatrix}.$$

8.3.3 Quadrature Rules

In order to derive the asynchronous integrator in Sect. 8.3.4, we have to be more specific about the synchronous integrator in terms of choosing the discrete Lagrangian (8.5). As introduced at the end of Sect. 8.3.1, the general coordinate q represents x and Λ and the partial derivatives $\frac{\partial(\cdot)}{\partial q}$ denote $\frac{\partial(\cdot)}{\partial x}$ and $2\left[\left(\Lambda^T \frac{\partial(\cdot)}{\partial \Lambda}\right)^{(A)}\right]^{\vee}$, respectively. Note that with regard to the asynchronous generalization, we denote the discrete Lagrangian in the element K_a^j by L_k^j (instead of L_a^j) from now on. Furthermore, time nodes associated with this element are denoted by t_k^j. As shown in (8.4) and (8.5), the discrete Lagrangian L_k^j is an approximation of the action integral over the space time element, and can generally depend on four nodes. As already mentioned in the beginning of Sect. 8.3, the specific choice of approximation, i.e. the quadrature used to approximate the integral of the Lagrangian density, determines the coupling structure and the properties of the resulting time stepping scheme. For example, in [18] approximations based on three and four mesh points are introduced, where the resulting numerical scheme based on four nodes is shown to be more stable compared to the three node approximation. In the following, we introduce a choice of approximation based on four nodes which is also particularly well suited for the generalization to asynchronous integrators. To this end, we partition the space time element K_a^j into four parts denoted by roman numerals as shown in Fig. 8.4. With $\Delta t = t_k^{j+1} - t_k^j = \text{const.}$ for all j, k and $\Delta s_k = s_{a+1} - s_a$, the time derivatives $\dot{x}_a^j$ and space derivatives x'^j_a are approximated by a forward difference quotient that takes the form

$$\dot{x}_a^j = \frac{x_a^{j+1} - x_a^j}{\Delta t} \quad x'^j_a = \frac{x_{a+1}^j - x_a^j}{\Delta s_k} \tag{8.16}$$

for the translational degrees of freedom. The temporal and spatial derivatives for the rotational degrees of freedom take a slightly more complicated form. Assuming a geodesic trajectory of the rotation during one time interval, i.e. $\Lambda_a^{j+1} = \Lambda_a^j e^{\Delta t \hat{\omega}_a^j}$ using the matrix exponential, the angular velocity ω in this time interval fulfills

$$\hat{\omega}_a^j = \frac{1}{\Delta t} \log\left(\left(\Lambda_a^j\right)^T \Lambda_a^{j+1}\right)$$

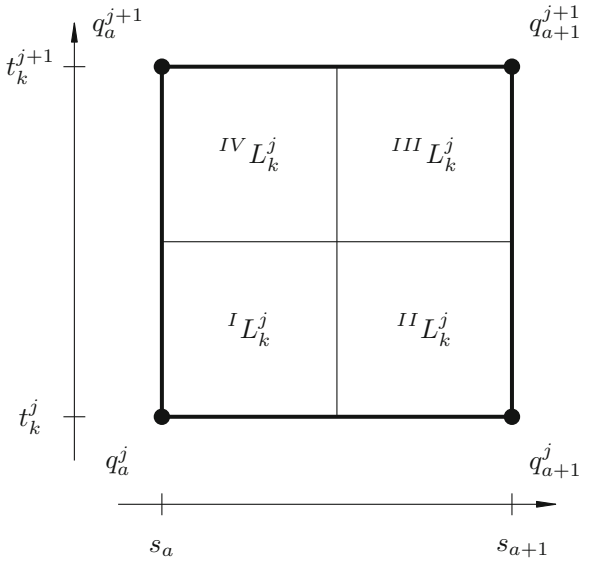


Fig. 8.4 Partitioning of the space time element K_a^j for the synchronous integrator and the general case of $L = L\left(q, \dot{q}, q'\right)$

where the logarithm of an orthonormal matrix with rotation angle $\varphi_a^j = \Delta t \left\|\omega_a^j\right\|$ can be written as

$$\log\left(\left(\Lambda_a^j\right)^T \Lambda_a^{j+1}\right) = \begin{cases} \dfrac{\varphi}{\sin\varphi_a^j}\left[\left(\Lambda_a^j\right)^T \Lambda_a^{j+1}\right]^{(A)} & \text{for } \varphi_a^j \in (-\pi, \pi) \setminus \{0\} \\ \hat{0} & \text{for } \varphi_a^j = 0 \end{cases}.$$

With $\lim\limits_{\varphi_a^j \to 0}\left(\varphi_a^j / \sin\varphi_a^j\right) = 1$, for small incremental rotations, the approximate angular velocity—and with a similar derivation the angular strain—are given by

$$\hat{\omega}_a^j = \frac{1}{\Delta t}\left[\left(\Lambda_a^j\right)^T \Lambda_a^{j+1}\right]^{(A)} \quad \hat{\Omega}_a^j = \frac{1}{\Delta s_k}\left[\left(\Lambda_a^j\right)^T \Lambda_{a+1}^j\right]^{(A)}. \tag{8.17}$$

The discrete Lagrangian is decomposed into

$$L_k^j\left(q_a^j, q_a^{j+1}, q_{a+1}^j, q_{a+1}^{j+1}\right) = {}^{I}L_k^j + {}^{II}L_k^j + {}^{III}L_k^j + {}^{IV}L_k^j$$

where the four parts consist of evaluations of the Lagrangian at different nodes and difference quotients[3]

$$
\begin{aligned}
{}^{IV}L_k^j &= \frac{\Delta t \Delta s_k}{4} L\left(q_a^{j+1}, \dot{q}_a^j, {q'}_a^{j+1}\right) & {}^{III}L_k^j &= \frac{\Delta t \Delta s_k}{4} L\left(q_{a+1}^{j+1}, \dot{q}_{a+1}^j, {q'}_a^{j+1}\right) \\
{}^{I}L_k^j &= \frac{\Delta t \Delta s_k}{4} L\left(q_a^j, \dot{q}_a^j, {q'}_a^j\right) & {}^{II}L_k^j &= \frac{\Delta t \Delta s_k}{4} L\left(q_{a+1}^j, \dot{q}_{a+1}^j, {q'}_a^j\right).
\end{aligned}
$$

The discrete Euler-Lagrange equations (8.7) for this specific quadrature are

$$
\begin{aligned}
\frac{\partial S_d}{\partial q_a^j} = & \frac{1}{4}\left(\frac{\partial {}^{I}L_k^j}{\partial q_a^j} + \frac{\partial {}^{II}L_k^j}{\partial q_a^j} + \frac{\partial {}^{IV}L_k^j}{\partial q_a^j}\right) + \frac{1}{4}\left(\frac{\partial {}^{I}L_k^{j-1}}{\partial q_a^j} + \frac{\partial {}^{III}L_k^{j-1}}{\partial q_a^j} + \frac{\partial {}^{IV}L_k^{j-1}}{\partial q_a^j}\right) \\
& + \frac{1}{4}\left(\frac{\partial {}^{I}L_{k-1}^j}{\partial q_a^j} + \frac{\partial {}^{II}L_{k-1}^j}{\partial q_a^j} + \frac{\partial {}^{III}L_{k-1}^j}{\partial q_a^j}\right) \\
& + \frac{1}{4}\left(\frac{\partial {}^{II}L_{k-1}^{j-1}}{\partial q_a^j} + \frac{\partial {}^{III}L_{k-1}^{j-1}}{\partial q_a^j} + \frac{\partial {}^{IV}L_{k-1}^{j-1}}{\partial q_a^j}\right) \\
= & \, 0.
\end{aligned} \tag{8.18}
$$

Quadrature rules of this type are visualized in Fig. 8.3. They lead to an integrator with independent sets of six non linear equations for each spatial node in one time step, since clearly (8.14) holds true. For the special case of $L\left(q, \dot{q}, q'\right) = T\left(q, \dot{q}\right) - V\left(q, q'\right)$, where T is the kinetic energy density and V is the potential energy density, the partitioning of the space time element is visualized in Fig. 8.5. The discrete kinetic energy densities are given by

$$
\begin{aligned}
{}^{II}_{k}T_a^j &= T\left(q_a^{j+1}, \dot{q}_a^j\right) & {}^{II}_{k}T_{a+1}^j &= T\left(q_{a+1}^{j+1}, \dot{q}_{a+1}^j\right) \\
{}^{I}_{k}T_a^j &= T\left(q_a^j, \dot{q}_a^j\right) & {}^{I}_{k}T_{a+1}^j &= T\left(q_{a+1}^j, \dot{q}_{a+1}^j\right)
\end{aligned} \tag{8.19}
$$

and the discrete potential energy densities read

$$
\begin{aligned}
{}^{IV}V_k^j &= V\left(q_a^{j+1}, {q'}_a^{j+1}\right) & {}^{III}V_k^j &= V\left(q_{a+1}^{j+1}, {q'}_a^{j+1}\right) \\
{}^{I}V_k^j &= V\left(q_a^j, {q'}_a^j\right) & {}^{II}V_k^j &= V\left(q_{a+1}^j, {q'}_a^j\right)
\end{aligned} \tag{8.20}
$$

yielding the discrete Lagrangian for this special case

$$
L_k^j = \frac{\Delta t \Delta s_k}{4}\left({}^{I}_{k}T_a^j + {}^{II}_{k}T_a^j + {}^{I}_{k}T_{a+1}^j + {}^{II}_{k}T_{a+1}^j - {}^{I}V_k^j - {}^{II}V_k^j - {}^{III}V_k^j - {}^{IV}V_k^j\right)
$$

[3] Of course, other evaluations of the Lagrangian depending on a different number or combinations of nodes are possible, as e.g. an evaluation at midpoints of nodes in the first argument of L (cf. [18]).

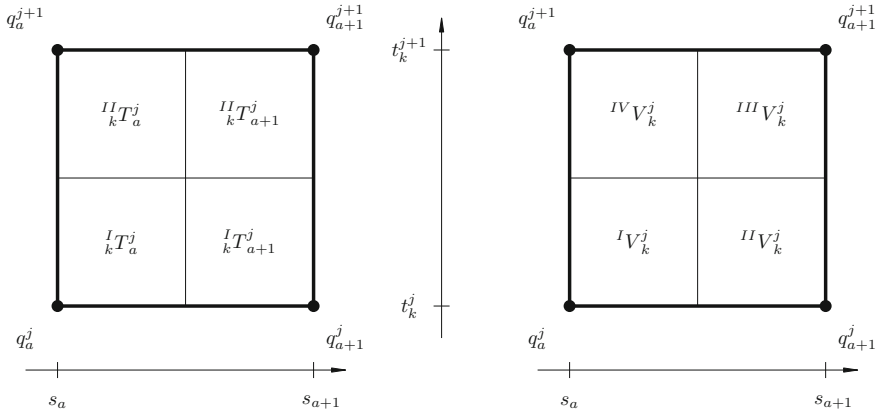


Fig. 8.5 Partitioning of the space time element K_a^j for the synchronous integrator and the special case of $L\left(q,\dot{q},q'\right)=T\left(q,\dot{q}\right)-V\left(q,q'\right)$

and the discrete Euler-Lagrange equations are

$$\begin{aligned}\frac{\partial S_d}{\partial q_a^j}=&\frac{1}{4}\left(\frac{\partial {}^{I}_{k}T^j_a}{\partial q_a^j}+\frac{\partial {}^{II}_{k}T^j_a}{\partial q_a^j}-\frac{\partial {}^{I}V^j_k}{\partial q_a^j}-\frac{\partial {}^{II}V^j_k}{\partial q_a^j}\right)\Delta t\,\Delta s_k\\&+\frac{1}{4}\left(\frac{\partial {}^{I}_{k}T^{j-1}_a}{\partial q_a^j}+\frac{\partial {}^{II}_{k}T^{j-1}_a}{\partial q_a^j}-\frac{\partial {}^{III}V^{j-1}_k}{\partial q_a^j}-\frac{\partial {}^{IV}V^{j-1}_k}{\partial q_a^j}\right)\Delta t\,\Delta s_k\\&+\frac{1}{4}\left(\frac{\partial {}^{I}_{k-1}T^{j}_{a-1}}{\partial q_a^j}+\frac{\partial {}^{II}_{k-1}T^{j}_{a-1}}{\partial q_a^j}-\frac{\partial {}^{I}V^{j}_{k-1}}{\partial q_a^j}-\frac{\partial {}^{II}V^{j}_{k-1}}{\partial q_a^j}\right)\Delta t\,\Delta s_{k-1}\\&+\frac{1}{4}\left(\frac{\partial {}^{I}_{k-1}T^{j-1}_{a-1}}{\partial q_a^j}+\frac{\partial {}^{II}_{k-1}T^{j-1}_{a-1}}{\partial q_a^j}-\frac{\partial {}^{III}V^{j-1}_{k-1}}{\partial q_a^j}-\frac{\partial {}^{IV}V^{j-1}_{k-1}}{\partial q_a^j}\right)\Delta t\,\Delta s_{k-1}=0.\end{aligned}\tag{8.21}$$

8.3.4 Asynchronous Generalization

As a generalization of the synchronous time integrator, we choose different time steps for each element which leads to the asynchronous space time grid shown in Fig. 8.6. The asynchronous partitioning of one space time element is shown in Fig. 8.7. The main consequence is, that in addition to elemental time nodes t_k^j, now there are also nodal time nodes t_a^i, e.g. for the spatial node s_a, the nodal time nodes are the union of the elemental time nodes of the adjacent element sharing the node s_a.

In order to define the discrete Lagrangian, we treat the kinetic energy density and the potential energy density in different ways. Consider a continuous Lagrangian of the form $L\left(q,\dot{q},q'\right)=T\left(q,\dot{q}\right)-V\left(q,q'\right)$. Space derivatives of the translational

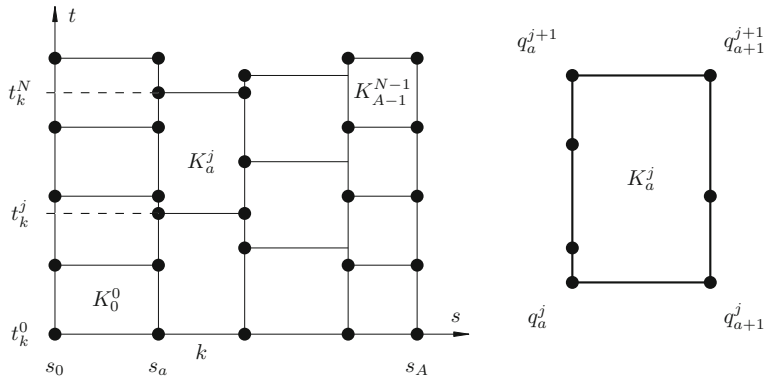


Fig. 8.6 Asynchronous discretization of space time for the asynchronous integrator

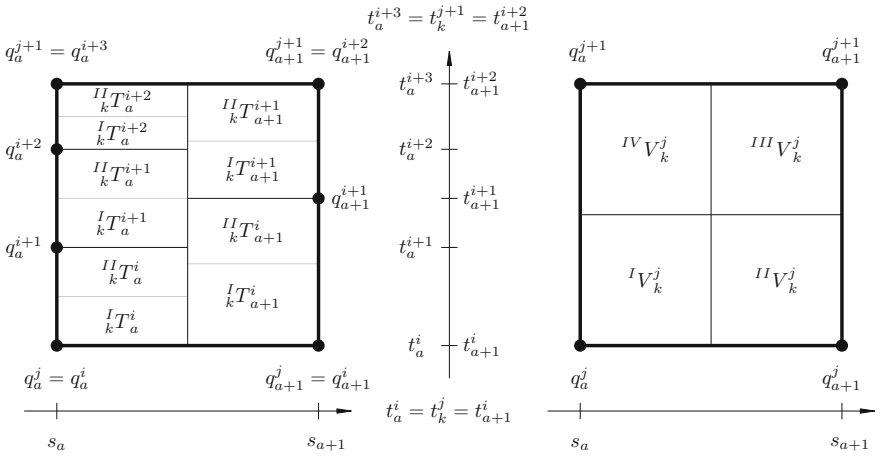


Fig. 8.7 Partitioning of the space time element K_a^j for the asynchronous integrator

and rotational degrees of freedom are approximated as given in (8.16) and (8.17), respectively. However, time derivatives are approximated by the forward difference quotient based on the nodal time step $\Delta t_a^i = t_a^{i+1} - t_a^i$ which has to be used in (8.16) and (8.17) together with nodal quantities x_a^i, x_a^{i+1} and Λ_a^i, Λ_a^{i+1}. Then, discrete kinetic energy densities are different from (8.19) since they are based on finite differences on the nodal time grid, namely

$$\begin{aligned} {}_k^{II}T_a^i &= T\left(q_a^{i+1}, \dot{q}_a^i\right) \quad {}_k^{II}T_{a+1}^i = T\left(q_{a+1}^{i+1}, \dot{q}_{a+1}^i\right) \\ {}_k^{I}T_a^i &= T\left(q_a^i, \dot{q}_a^i\right) \quad {}_k^{I}T_{a+1}^i = T\left(q_{a+1}^i, \dot{q}_{a+1}^i\right) \end{aligned} \tag{8.22}$$

while the discrete potential energy densities are equal to (8.20) defined on the elemental time grid. In terms of the elemental time step $\Delta t_k = t_k^{j+1} - t_k^j = \text{const.}$ in j, the discrete Lagrangian of the element takes the form

$$\begin{aligned} L_k^j &= \sum_{a=k}^{k+1} \sum_{\left\{i \,|\, t_a^j \le t_a^i < t_a^{j+1}\right\}} \frac{\Delta t_a^i \Delta s_k}{4} \left({}^{I}_{k}T_a^i + {}^{II}_{k}T_a^i \right) \\ &\quad - \frac{\Delta t_k \Delta s_k}{4} \left({}^{I}V_k^j + {}^{II}V_k^j + {}^{III}V_k^j + {}^{IV}V_k^j \right). \end{aligned} \tag{8.23}$$

The discrete Euler-Lagrange equations (8.7) for this specific asynchronous quadrature are

$$\begin{aligned} \frac{\partial S_d}{\partial q_a^i} &= \frac{1}{4}\left(\frac{\partial {}^{I}_{k}T_a^i}{\partial q_a^i} + \frac{\partial {}^{II}_{k}T_a^i}{\partial q_a^i} \right) \Delta t_a^i \Delta s_k \quad - \frac{1}{4}\left(\frac{\partial {}^{I}V_k^j}{\partial q_a^j} + \frac{\partial {}^{II}V_k^j}{\partial q_a^j} \right) \Delta t_k \Delta s_k \Big|_{t_a^i == t_k^j} \\ &+ \frac{1}{4}\left(\frac{\partial {}^{I}_{k}T_a^{i-1}}{\partial q_a^i} + \frac{\partial {}^{II}_{k}T_a^{i-1}}{\partial q_a^i} \right) \Delta t_a^{i-1} \Delta s_k \quad - \frac{1}{4}\left(\frac{\partial {}^{IV}V_k^{j-1}}{\partial q_a^j} + \frac{\partial {}^{III}V_k^{j-1}}{\partial q_a^j} \right) \Delta t_k \Delta s_k \Big|_{t_a^i == t_k^j} \\ &+ \frac{1}{4}\left(\frac{\partial {}^{I}_{k-1}T_{a-1}^i}{\partial q_a^i} + \frac{\partial {}^{II}_{k-1}T_{a-1}^i}{\partial q_a^i} \right) \Delta t_a^i \Delta s_{k-1} \quad - \frac{1}{4}\left(\frac{\partial {}^{I}V_{k-1}^j}{\partial q_a^j} + \frac{\partial {}^{II}V_{k-1}^j}{\partial q_a^j} \right) \Delta t_{k-1} \Delta s_{k-1} \Big|_{t_a^i == t_{k-1}^j} \\ &+ \frac{1}{4}\left(\frac{\partial {}^{I}_{k-1}T_{a-1}^{i-1}}{\partial q_a^i} + \frac{\partial {}^{II}_{k-1}T_{a-1}^{i-1}}{\partial q_a^i} \right) \Delta t_a^{i-1} \Delta s_{k-1} \quad - \frac{1}{4}\left(\frac{\partial {}^{IV}V_{k-1}^{j-1}}{\partial q_a^j} + \frac{\partial {}^{III}V_{k-1}^{j-1}}{\partial q_a^j} \right) \Delta t_{k-1} \Delta s_{k-1} \Big|_{t_a^i == t_{k-1}^j} \\ &= 0. \end{aligned} \tag{8.24}$$

The expression $|_{t_a^i == t_k^j}$ is to be read as a condition, i.e. if the nodal time t_a^i is equal to the elemental time t_k^j, the element to the right influences the node by contributing its potential term into the balance of momentum–if $t_a^i \neq t_k^j$, the potential term is zero. Similarly, for $|_{t_a^i == t_{k-1}^j}$ the left element influences the node. The Eq. (8.24) are a generalization of the synchronous integrator since for $\Delta t_a^i = \Delta t_k = \Delta t = \text{const.}$ in $i = j, k$, Eq. (8.21) are recovered.

8.4 Geometrically Exact Beam Dynamics

Modeling geometrically exact beams as a special Cosserat continuum (see e.g., [1]) has been the basis for many discrete formulations starting with [37, 53–55]. Its Lagrangian dynamics is an example of the formulation described in Sect. 8.2.2.

8.4.1 Continuous Euler-Lagrange Equations

The beam model consists of a central line $x(s,t)$, a curve in the three dimensional space, along which rotation matrices $\Lambda(s,t)$ represent the orientations of the local beam cross sections. Consequently, the cross sections are assumed to stay plane. Hereby, the curve parameter is $s \in [0, \ell]$, where ℓ denotes the length of the beam in the reference configuration. As mentioned in Sect. 8.2.2, in addition to the configuration, the Lagrange density is a function of the angular velocity $\hat{\omega} = \Lambda^T \dot{\Lambda} \in \mathfrak{so}(3)$ and bending and torsional strain $\hat{\Omega} = \Lambda^T \Lambda' \in \mathfrak{so}(3)$ as well as the translational velocity $\dot{x}$ and the shear and elongational strain x'

$$\begin{aligned} L\left(\Lambda, \omega, \Omega, x, \dot{x}, x'\right) = & \frac{1}{2}\left(\rho \left\|\dot{x}\right\|^2 + \omega^T J \omega\right) \\ & - \frac{1}{2}\left[\left(\Lambda^T x' - e_3\right)^T C_1 \left(\Lambda^T x' - \boldsymbol{e}_3\right) + \Omega^T C_2 \Omega\right] \\ & + \rho \langle x, g \rangle . \end{aligned} \tag{8.25}$$

Here, the kinetic energy density contains the mass density ρ, the mass moment of inertia density tensor J, while the gravity potential energy density depends on the gravity constant g. In the potential deformation energy density, the diagonal matrix $C_1 = \mathrm{Diag}\left(GA \quad GA \quad EA\right)$ contains the shear and elongation stiffness, while the entries of $C_2 = \mathrm{Diag}\left(EI_1 \quad EI_2 \quad G\left(I_1 + I_2\right)\right)$ are the bending and torsional stiffness. Here, A is the cross section area, I_1, I_2 are the principal area moments of inertia of the cross section, and the elastic properties are represented by Young's modulus E and the shear modulus G. Insertion into (8.1) yields the Euler-Lagrange equations for the beam

$$\begin{aligned} \left(\Lambda^T x'\right) \times \left(C_1 \left(\Lambda^T x' - \boldsymbol{e}_3\right)\right) - \omega \times (J\omega) - J\dot{\omega} + \Omega \times (C_2 \Omega) + C_2 \Omega' = 0 \\ -\rho g + \rho \ddot{x} + \Lambda \left(\Omega \times C_1 \left(\Lambda^T x' - \boldsymbol{e}_3\right)\right) + \Lambda C_1 \left(\left(\Lambda'\right)^T x' + x''\right) = 0. \end{aligned}$$

The conjugate momenta (8.2) are given by

$$\begin{aligned} \Pi = \frac{\partial L}{\partial \omega} = J\omega \qquad \Gamma = \frac{\partial L}{\partial \dot{x}} = \rho \dot{x} \\ \Sigma = \frac{\partial L}{\partial \Omega} = -C_2 \Omega \quad \sigma = \frac{\partial L}{\partial x'} = -\Lambda C_1 \left(\Lambda^T x' - \boldsymbol{e}_3\right). \end{aligned}$$

They can be identified as angular momentum per length Π, linear momentum per length Γ, and Σ represents the bending and torsional momenta while σ comprises shear and elongation forces.[4] In case of the beam, in addition to prescribing the

[4] The unit of the Lagrangian density is $\frac{\mathrm{J}}{\mathrm{m}} = N$, i.e. energy per length. Therefore, the units for the conjugate momenta are

Table 8.1 Simulation parameters

Length	2 m
Cross section	0.01 m × 0.01 m
Density	$1{,}000\,\mathrm{kg/m^3}$
Young's modulus	$5 \times 10^6\,\mathrm{N/m^2}$
Poisson ratio	0.35
Number of elements	11
Spatial discretization	$\Delta s_k = \frac{2}{11}\mathrm{m}$
Time steps sync	$\Delta t = 4 \times 10^{-4}\mathrm{s}$
Time steps async	Δt_k randomly in $[3, 4] \times 10^{-4}\,\mathrm{s}$
Simulation time	10 s

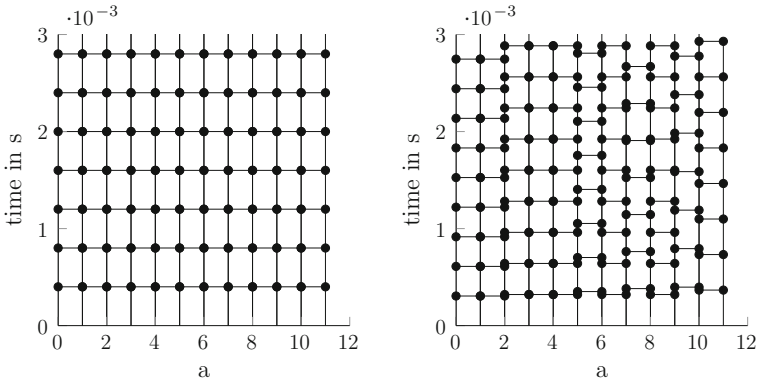


Fig. 8.8 Space time grid for the synchronous (*left*) and asynchronous (*right*) simulation of the beam for the first 3×10^{-3}s

position or orientation at specific nodes, further boundary conditions in space are the momenta Σ_0, Σ_A and forces σ_0, σ_A on the end nodes of the beam. Further boundary conditions in time are the angular momenta Π_0, Π_N in the beginning and the end of time, as well as linear momenta Γ_0, Γ_N. These can be inserted in (8.12) and (8.13) to obtain boundary conditions for the discrete Euler-Lagrange equations (8.11).

(Footnote 4 continued)

$$[\Pi] = \frac{\mathrm{Ns}}{\mathrm{m}} = \frac{\frac{\mathrm{kgm}^2}{\mathrm{s}}}{\mathrm{m}} \qquad [\Gamma] = \frac{\mathrm{Ns}}{\mathrm{m}} = \frac{\frac{\mathrm{kgm}}{\mathrm{s}}}{\mathrm{m}} \qquad [\Sigma] = \mathrm{Nm} \qquad [\sigma] = \mathrm{N}.$$

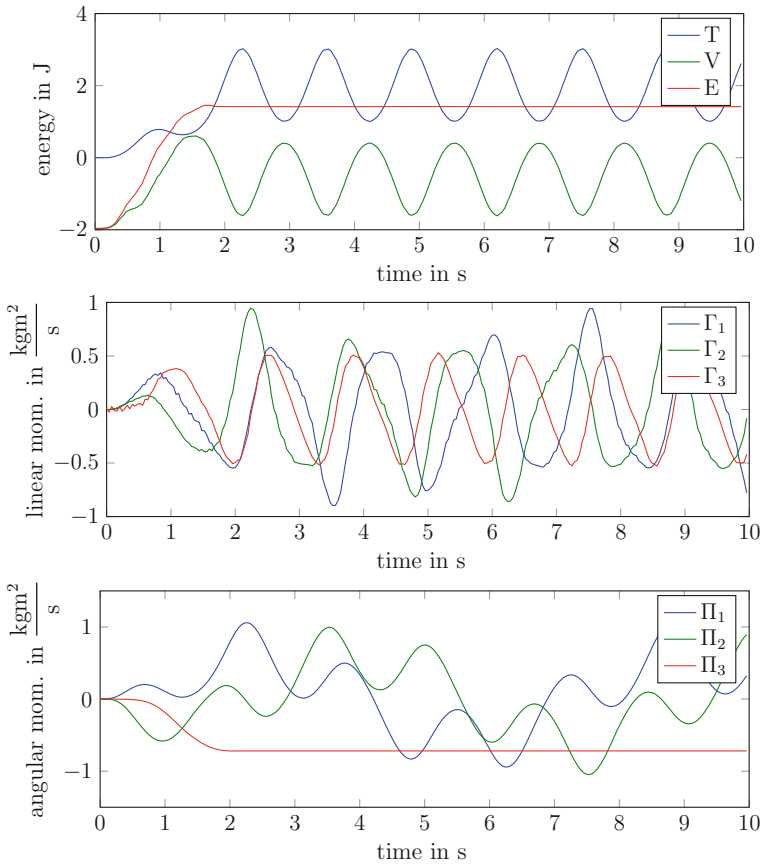


Fig. 8.9 Energy, linear and angular momenta for the synchronous simulation of the beam

8.4.2 Asynchronous Discretization

The approximation of time and space derivatives are performed as described in Sects. 8.3.3 and 8.3.4. Insertion into the discrete kinetic energy densities (8.22) and in the discrete potential energy densities (8.20) leads to the discrete Lagrangian (8.23). Specifically, for the Lagrangian of the beam (8.25), the discrete Lagrangian is composed by the discrete kinetic energy densities

$$ {}^{I}_{k}T^{i}_{a} = {}^{II}_{k}T^{i}_{a} = \frac{1}{2}\left(\rho_k \left\|\dot{x}^{i}_{a}\right\|^2 + \left(\omega^{i}_{a}\right)^T J \omega^{i}_{a}\right) $$

and the discrete potential energy densities

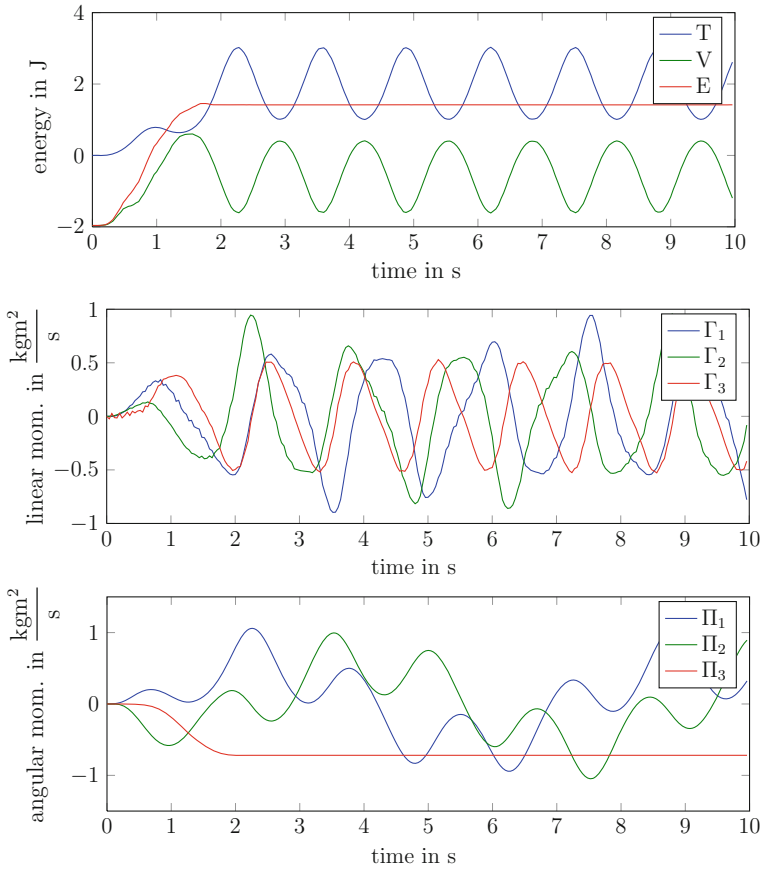


Fig. 8.10 Energy, linear and angular momenta for the asynchronous simulation of the beam

$$
\begin{aligned}
{}^{I}V_k^j &= \frac{1}{2}\left[\left(\left(\Lambda_a^j\right)^T x'^j_a - \boldsymbol{e}_3\right)^T C_1 \left(\left(\Lambda_a^j\right)^T x'^j_a - \boldsymbol{e}_3\right) + \left(\Omega_a^j\right)^T C_2 \Omega_a^j\right] + \rho_k \left\langle x_a^j, g\right\rangle \\
{}^{II}V_k^j &= \frac{1}{2}\left[\left(\left(\Lambda_{a+1}^j\right)^T x'^j_a - \boldsymbol{e}_3\right)^T C_1 \left(\left(\Lambda_{a+1}^j\right)^T x'^j_a - \boldsymbol{e}_3\right) + \left(\Omega_a^j\right)^T C_2 \Omega_a^j\right] + \rho_k \left\langle x_{a+1}^j, g\right\rangle \\
{}^{III}V_k^j &= \frac{1}{2}\left[\left(\left(\Lambda_{a+1}^{j+1}\right)^T x'^{j+1}_a - \boldsymbol{e}_3\right)^T C_1 \left(\left(\Lambda_{a+1}^{j+1}\right)^T x'^{j+1}_a - \boldsymbol{e}_3\right) + \left(\Omega_a^{j+1}\right)^T C_2 \Omega_a^{j+1}\right] + \rho_k \left\langle x_{a+1}^{j+1}, g\right\rangle \\
{}^{IV}V_k^j &= \frac{1}{2}\left[\left(\left(\Lambda_a^{j+1}\right)^T x'^{j+1}_a - \boldsymbol{e}_3\right)^T C_1 \left(\left(\Lambda_a^{j+1}\right)^T x'^{j+1}_a - \boldsymbol{e}_3\right) + \left(\Omega_a^{j+1}\right)^T C_2 \Omega_a^{j+1}\right] + \rho_k \left\langle x_a^{j+1}, g\right\rangle.
\end{aligned}
$$

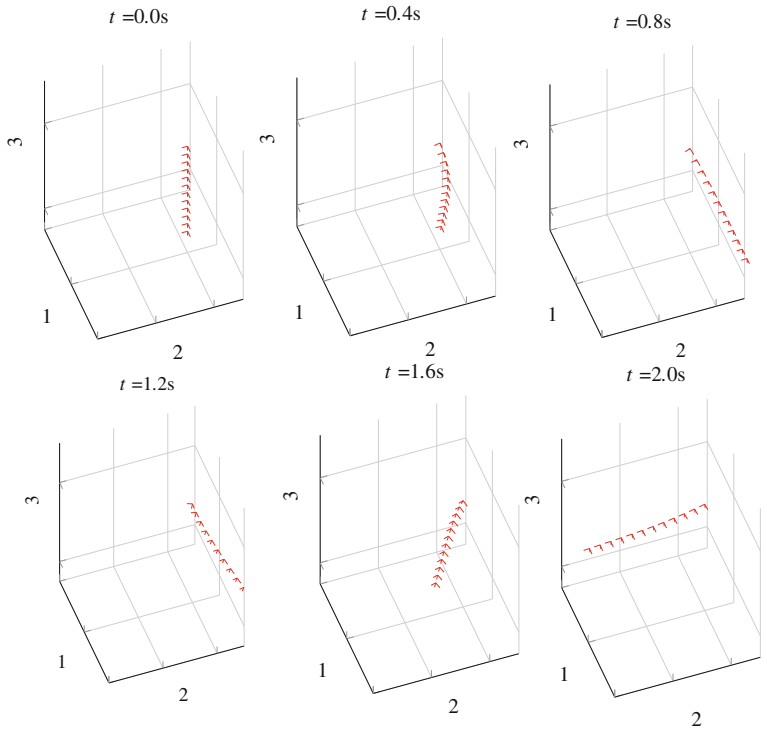


Fig. 8.11 Snapshots of the beam configuration (the three directors forming the columns of the orientation matrix Λ_a^j are visualized) at different times

8.5 Numerical Results

The AVI is implemented in Matlab® using the discretization discussed above. The discrete Euler-Lagrange equations are derived using Matlab®'s symbolic toolbox and automatic code generation. The derivations of the discrete Euler-Lagrange equations (8.24) are performed with the specific goal in mind to allow the automatic generation of AVI code. The procedure also allows the use of automatic differentiation, see e.g. [56]. This enables very fast prototyping of integration methods based on the discrete variational principle, e.g. for testing new models.

In order to test the AVI a beam with the physical parameters shown in Table 8.1 is discretized according to Sect. 8.4.2. The first node of the beam is translationally fixed and the initial configuration consists of the beam hanging downward (in the negative z-direction). During the first two seconds of the simulation time, a torque is applied to the first node. During that timeframe, the energy and the linear and angular momenta of the beam are expected to change, while after two seconds we expect the total energy to be nearly constant and the e_3-component of the angular momentum Π_3^j to be exactly preserved, since the Lagrangian is invariant with respect

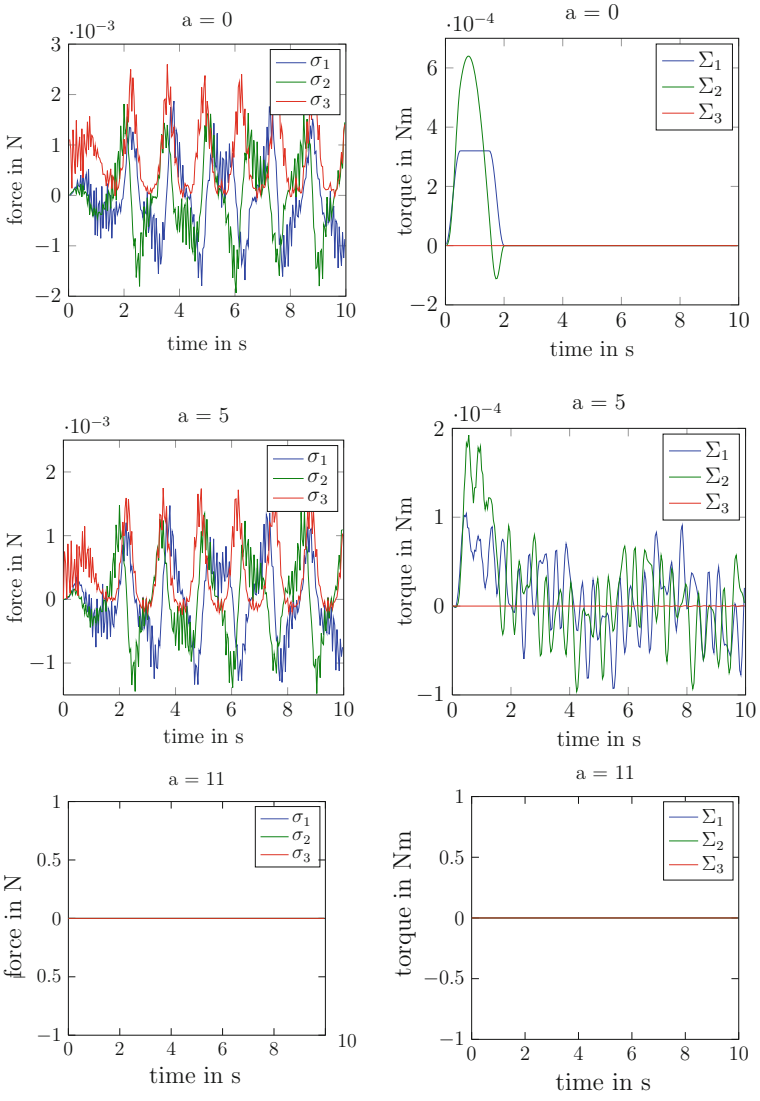


Fig. 8.12 Forces and torques as stress resultants at nodes 0, 5 and 11

to time and rotations around the e_3-axis. The simulation is done synchronously with $\Delta t = 3 \times 10^{-3}$s and asynchronously with time steps randomly distributed between 3×10^{-3}s and 4×10^{-3}s. The space time grids are shown in Fig. 8.8. Figures 8.9 and 8.10 illustrate the expected behavior in the plots for the energy and the linear and angular momenta, while Fig. 8.11 presents snapshots of the configuration of the beam at different times. Figure 8.12 shows the evolution of the conjugate momenta

at the nodes $a = 0$, $a = 5$ and $a = 11$. The torque Σ at node $a = 0$ is the actuation that decays after 2 s. Afterwards, the orientation at this node is free and thus the reaction momentum is zero. However, the position of node $a = 0$ is fixed, thus, the reaction force is non zero. This is in contrast to node $a = 11$, since at the free end of the beam, reaction forces and momenta are always zero.

8.6 Conclusions and Outlook

In this work, a variational integrator for Lagrangian PDE systems defined on the Lie group $SO(3)$ is derived based on a discrete action on a grid in space and time. By allowing only group operations when varying the discrete action and updating the group elements, the resulting numerical scheme naturally preserves the Lie group structure of the Lagrangian system. The discrete formulation of the Lagrangian system allows the construction of synchronous as well as asynchronous numerical integration schemes which involves the use of varying time step sizes throughout different elements in space. The presented variational Lie group integrator is applied for the synchronous and asynchronous simulation for geometrically exact beam dynamics formulated on $SO(3)$ and corresponding numerical results are presented. The exact preservation of angular and linear momentum is illustrated in the numerical example. This is due to the symplectic nature of the integrator, which is also responsible for the good energy behavior. In future works, the advantage in computing times for the asynchronous integrator has to be verified by taking varying material and geometrical parameters throughout the beam into account. In this case, using an AVI prevents the stiffer regions from dictating small time steps for the whole beam.

References

1. Antmann SS (1995) Nonlinear problems in elasticity. Springer
2. Lew A, Marsden JE, Ortiz M, West M (2003) Asynchronous variational integrators. Arch Ration Mech Anal 167(2):85–146
3. Lee T, Leok M, McClamroch NH (2007) Lie group variational integrators for the full body problem. Comput Methods Appl Mech Eng 196(29–30):2907–2924
4. Hairer E, Lubich C, Wanner G (2002) Geometric numerical integration, volume 31 of Springer series in computational mathematics. Springer
5. Marsden JE, West M (2001) Discrete mechanics and variational integrators. Acta Numer 10:357–514
6. Lew A, Marsden JE, Ortiz M, West M (2004) An overview of variational integrators. Finite element methods: 1970's and beyond, pp 98–115
7. Lew A, Marsden JE, Ortiz M, West M (2004) Variational time integrators. Int J Numer Methods Eng 60(1):153–212
8. Ober-Blöbaum S, Junge O, Marsden JE (2008) Discrete mechanics and optimal control: an analysis. ESAIM: Control Optim Calc Var 17(2):322–352. arXiv:0810.1386

9. Kane C, Marsden JE, Ortiz M, West M (1999) Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. Int J Numer Methods Eng 49(10):1295–1325
10. Leyendecker S, Marsden JE, Ortiz M (2008) Variational integrators for constrained dynamical systems. ZAMM: J Appl Math Mech/Zeitschrift für Angewandte Mathematik und Mechanik 88(9):677–708
11. Leyendecker S, Ober-Blöbaum S, Marsden JE, Ortiz M (2010) Discrete mechanics and optimal control for constrained systems. Optimal Control Appl Methods 31(6):505–528
12. Kobilarov M, Marsden JE, Sukhatme GS (2010) Geometric discretization of nonholonomic systems with symmetries. Discrete Continuous Dyn Syst: Series S 1(1):61–84
13. Fetecau RC, Marsden JE (2003) Nonsmooth Lagrangian mechanics and variational collision integrators. SIAM J Appl Dyn Syst 3(2):381–416
14. Bou-Rabee N, Owhadi H (2008) Stochastic variational integrators. IMA J Numer Anal 29(2):421–443
15. Tao M, Owhadi H, Marsden JE (2010) Non-intrusive and structure preserving multiscale integration of stiff ODEs, SDEs and Hamiltonian systems with hidden slow dynamics via flow averaging. Multiscale Model Simul 8(4):1269–1324
16. Leyendecker S, Ober-Blöbaum S (2013) A variational approach to multirate integration for constrained systems. Comput Methods Appl Sci 28:97–121
17. Ober-Blöbaum S, Tao M, Cheng M, Owhadi H, Marsden JE (2011) Variational integrators for electric circuits. J Comput Phys 242:498–530
18. Marsden JE, Patrick GW, Shkoller S (1998) Multisymplectic geometry, variational integrators, and nonlinear PDEs. Commun Math Phys 199:351–395
19. Vankerschaver J, Liao C, Leok M (2011) Generating functionals and Lagrangian PDEs. J Nonlinear Sci (submitted, Marsden memorial issue, invited paper, arXiv:1111.0280)
20. Kale KG, Lew AJ (2007) Parallel asynchronous variational integrators. Int J Numer Methods Eng 70:291–321
21. Focardi M, Mariano PM (2008) Convergence of asynchronous variational integrators in linear elastodynamics. Int J Numer Meth Eng 75(7):755–769
22. Beneš M, Matoušlew K (2010) Asynchronous multi-domain variational integrators for nonlinear hyperelastic solids. Comput Methods Appl Mech Eng 199(29–32):1992–2013
23. Vouga E, Harmon D, Tamstorf R, Grinspun E (2010) Asynchronous variational contact mechanics. arXiv:1007.3240
24. Wolff S, Bucher C (2013) Asynchronous variational integration using continuous assumed gradient elements. Comput Methods Appl Mech Eng 255(C):158–166
25. Crouch PE, Grossman R (1993) Numerical integration of ordinary differential equations on manifolds. J Nonlinear Sci 3(1):1–33
26. Munthe-Kaas H (1998) Runge-Kutta methods on Lie groups. BIT Numer Math 38(1):92–111
27. Iserles A, Munthe-Kaas HZ, Nørsett SP, Zanna A (2000) Lie-group methods. Acta Numerica 9:215–365
28. Bobenko AI, Suris YB (1999) Discrete Lagrangian reduction, discrete Euler-Poincaré equations, and semidirect products. Lett Math Phys 49(1):79–93
29. Bobenko AI, Suris YB (1999) Discrete time Lagrangian mechanics on Lie groups, with an application to the Lagrange top. Commun Math Phys 204:147–188
30. Marsden JE, Pekarsky S, Shkoller S (1999) Discrete Euler-Poincaré and Lie-Poisson equations. Nonlinearity 12:1647–1662
31. Lee T (2008) Computational geometric mechanics and control of rigid bodies. PhD thesis, University of Michigan
32. Bou-Rabee N, Marsden JE (2009) Hamilton-Pontryagin integrators on Lie groups: introduction and structure-preserving properties. Found Comput Math 9(2):197–219
33. Celledoni E, Marthinsen H, Owren B (2014) An introduction to Lie group integrators—basics, new developments and applications. J Comput Phys 257:1040–1061
34. Lee T, Leok M, McClamroch NH (2007) Lie group variational integrators for the full body problem in orbital mechanics. Celest Mech Dyn Astron 98(2):121–144

35. Demoures F (2012) Lie group and Lie algebra variational integrators for flexible beam and plate in $\mathbb{R}^3$. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne
36. Demoures F, Gay-Balmaz F, Leitz T, Leyendecker S, Ober-Blöbaum S, Ratiu TS (2013) Asynchronous variational lie group integration for geometrically exact beam dynamics. In: ECCOMAS Thematic conference on mutlibody, Dynamics, 1–4 July 2013
37. Simo JC (1985) A finite strain beam formulation. the three-dimensional dynamic problem. Part I. Comput Methods Appl Mech Eng 49(1):55–70
38. Jelenić G, Crisfield MA (1998) Interpolation of rotational variables in non-linear dynamics of 3d beams. Int J Numer Methods Eng 43:1193–1222
39. Ibrahimbegović A, Mamouri S (1998) Finite rotations in dynamics of beams and implicit time-stepping schemes. Int J Numer Methods Eng 41:781–814
40. Crisfield MA, Jelenić G (1999) Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation. Proc Royal Soc London. Series A: Math Phys Eng Sci 455(1983):1125–1147
41. Romero I, Armero F (2002) An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energymomentum conserving scheme in dynamics. Int J Numer Methods Eng 54(12):1683–1716
42. Betsch P, Steinmann P (2002) Frame-indifferent beam finite elements based upon the geometrically exact beam theory. Int J Numerical Meth Eng 54:1775–1788
43. Romero I (2004) The interpolation of rotations and its application to finite element models of geometrically exact rods. Comput Mech 34(2):121–133
44. Betsch P, Menzel A, Stein E (1998) On the parametrization of finite rotations in computational mechanics: a classification of concepts with application to smooth shells. Comput Methods Appl Mech Eng 155:273–305
45. Ibrahimbegović A, Frey F, Kozar I (1995) Computational aspects of vector-like parametrization of three-dimensional finite rotations. Int J Numer Methods Eng 38:3653–3673
46. Jelenić G, Crisfield MA (1999) Geometrically exact 3d beam theory: implementation of a strain-invariant finite element for statics and dynamics. Comput Methods Appl Mech Eng 171:141–171
47. Jelenić G, Crisfield MA (2002) Problems associated with the use of Cayley transform and tangent scaling for conserving energy and momenta in the Reissner-Simo beam theory. Commun Numer Methods Eng 18:711–720
48. Bottasso CL, Borri M, Trainelli L (2002) Geometric invariance. Comput Mech 29:163–169
49. Shabana A (1998) Dynamics of multibody systems. Cambridge University Press
50. Shabana A, Yakoub RY (2001) Three dimensional absolute nodal coordinate formulation for beam elements: theory. ASME J Mech Des 123:606–613
51. Brüls O, Cardona A (2010) On the use of Lie group time integrators in multibody dynamics. J Comput Nonlinear Dyn 5
52. Brüls O, Cardona A, Arnold M (2012) Lie group generalized-α time integration of constrained flexible multibody systems. Mech Mach Theory 48:121–137
53. Simo JC, Vu-Quoc L (1986) A three-dimensional finite-strain rod model. Part II: computational aspects. Comput Methods Appl Mech Eng 58:79–116
54. Simo JC, Vu-Quoc L (1988) On the dynamics in space of rods undergoing large motions—a geometrically exact approach. Comput Methods Appl Mech Eng 66:125–161
55. Simo JC, Marsden JE, Krishnaprasad PS (1987) The Hamiltonian structure of nonlinear elasticity: the material and convective representations of solids, rods, and plates. Arch Ration Mech Anal 104:125–183
56. Walther A, Kowarz A, Griewank A (1996) ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. Assoc Comput Mach Trans Math Softw (ACM TOMS) 22(2):131–167

Chapter 9
On the Use of Geometrically Exact Shells for Dynamic Tire Simulation

Michael Roller, Peter Betsch, Axel Gallrein and Joachim Linn

Abstract In the present work a tire is modeled by using geometrically exact shells. The discretization is done with the help of isoparametric quadrilateral finite elements. The interpolation is performed with linear Lagrangian polynomials for the midsurface as well as for the director field. As time stepping method for the resulting differential algebraic equation a modified backward differential formula rule is chosen. To handle the interaction with a rigid road surface, a one sided normal contact formulation is introduced. An orthotropic material model for geometrically exact shells derived from 3D continuum theory is introduced, to describe the anisotropic behavior of the tire material. Inflation pressure is taken into account with a configuration dependent force. The interaction between the multibody system of a car and the tire is realized via co-simulation. Some quasi-static simulations are presented and compared to measurements on a real tire.

9.1 Introduction

Acting as an interface between car and road, the tire model plays an outstanding role in dynamic simulations of vehicles. In commercial and scientific application contexts there exist several different modeling approaches for tires, depending on the

M. Roller · A. Gallrein · J. Linn (✉)
Fraunhofer ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
e-mail: Joachim.Linn@itwm.fraunhofer.de

M. Roller
e-mail: Michael.Roller@itwm.fraunhofer.de

A. Gallrein
e-mail: Axel.Gallrein@itwm.fraunhofer.de

P. Betsch
Institute of Mechanics, Karlsruhe Institute of Technology (KIT), Otto-Ammann-Platz 9, 76131 Karlsruhe, Germany
e-mail: Peter.Betsch@kit.edu

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_9,

use case. When the tire model has to be embedded into a multi body system (MBS) of a car, lumped parameter models of varying complexity consisting of springs and dampers [14] are used, as well as simple data curve fits [15]. Likewise, very detailed but computationally demanding three dimensional finite element (FE) models are used for crash and misuse simulations [16]. However, a coupling of such 3D-FE tire models to MBS simulations is mostly not feasible—neither directly, nor via co-simulation—due to the large number of degrees of freedom (DOF).

Nevertheless, it would be highly desirable to have a continuum mechanics based structural model for tires available as an alternative with a moderate number of DOF, compatible with a direct usage in MBS simulations, and the essential structural properties of a tire properly incorporated, in order to be useful for practical applications. From the viewpoint of structural modeling, *geometrically exact shells* [17], which according to our knowledge have not yet been used in tire modeling, are a good candidate: This approach enables large rigid body motions, because stresses are constitutively related only to frame-indifferent (*objective*) differential invariants of the directed shell surface measuring membrane, bending and transverse shear strains on the deformed structure.

As mentioned above, the tire model should be incorporated into a multibody system. However, flexible multibody dynamics with geometrically exact shells have been discussed in a few works [1, 2, 13] only. By modeling the rim as a rigid body and fixing part of the shell boundary on it, we already obtain a minimal flexible multibody system, which could be built into a simple MBS model like a quarter vehicle, or further extended by coupling to a more complex MBS model of e.g. the axle construction of the car. In principle, four copies of such a shell based tire model could likewise be coupled to a MBS model of the full car in the same way, with an obvious approach for a proper parallelization of the resulting complex flexible MBS model in order to make it computationally feasible.

To model a tire on the basis of a geometrically exact shell structure, a variety of features of the real physical tire have to be included: The most important task of a tire is its role as the interface for the dynamic interaction of the vehicle and the road. Assuming the latter as *rigid*, the shell model has to be able to handle *one-sided dynamic contact* with a rigid road surface. Because of the strong reinforcements embedded within the rubber matrix, an isotropic material law, as often used within academic benchmark examples, is obviously *not* sufficient for realistic tire simulations. For this purpose, at least a more complex *orthotropic* material model has to be be chosen and properly transferred to the shell structure. To carry and absorb high static and dynamic loads, a tire is supported by *inflation pressure*, which leads to *configuration dependent forces*, acting as *live loads* on the shell structure, and has to be incorporated properly into our shell based tire model as well.

A brief outline of our article may be given as follows: Sect. 9.2 illustrates the continuum mechanical theory behind the geometrically exact shell model. Details on the constitutive model (as mentioned above) are presented in Sect. 9.3. Section 9.4 deals with the spatial and temporal discretization of the shell and the size reduction

of the resulting system of equations. The coupling between a multibody system and the discrete shell is topic of Sect. 9.5. Our approach to model the application of tire specific loads, especially pressure and normal contact, is explained in Sect. 9.6, and Sect. 9.7 contains some numerical results, with a focus on quasi-static test examples, including a comparison to experiments with a real tire, where the afore mentioned effects are acting in combination. In Sect. 9.8, we finish with a conclusion and an outlook of our further work.

9.2 Continuum Mechanical Formulation of the Shell Structure

Defining the shell continuum and its kinematics are the topics of this section. To fix ideas and notation, we briefly outline the dimensional reduction of the problem from the 3D continuum to the 2D shell structure, following the *geometrically exact* approach described by Simo and Fox [17].

9.2.1 Basic Definitions and Notation

A shell is a three dimensional continuum, whose extension h in thickness direction is *small* compared to its linear dimensions in the two other directions. A first approximation of the geometry of such a shell continuum is provided by its two dimensional *mid surface*, parametrized by the regular, vector valued function [7]

$$\varphi : \omega \ \rightarrow \ \mathbb{R}^3 \quad \omega \subset \mathbb{R}^2. \tag{9.1}$$

The derivatives of this mapping with respect to the variables χ_α of the parameter domain ω defines the *covariant* basis vectors[1]

$$\mathbf{a}_\alpha := \frac{\partial}{\partial \chi_\alpha} \varphi. \tag{9.2}$$

Due to the assumed regularity of the parametrization, the pair of vectors $\mathbf{a}_\alpha$ is linearily independent and spans the tangential plane of the surface in every point. The *mid surface normal* $\mathbf{a}_3 := \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|}$ completes $\mathbf{a}_i$ to a three dimensional basis of Euclidian

[1] Throughout this article, we follow the frequently used convention of denoting indices taking integer values 1 and 2 by lower case Greek letters $\alpha, \beta, \ldots$, while we use lowercase Latin ones $i, j, \ldots$ for integer indices ranging from 1 to 3, and we use *Einstein's summation convention* to abbreviate sums, ranging from 1 to 2 or 1 to 3 respectively, over (product type) terms with the same indices appearing twice.

space. The system of equations $\mathbf{a}_\alpha \cdot \mathbf{a}^\beta = \delta_\beta^\alpha$ uniquely determines the *contravariant* basis vectors $\mathbf{a}^\beta$, with $\mathbf{a}^3 = \mathbf{a}_3$ being self-dual, and the scalar products

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \quad \text{and} \quad a^{\alpha\beta} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta \tag{9.3}$$

yield the co- and contravariant components of the metric of the mid surface, induced by the scalar product in three-dimensional Euclidian space [7], such that the metric may be expanded as $\mathsf{a} = a_{\alpha\beta}\, \mathbf{a}^\alpha \otimes \mathbf{a}^\beta = a^{\alpha\beta}\, \mathbf{a}_\alpha \otimes \mathbf{a}_\beta = \mathbf{a}_\alpha \otimes \mathbf{a}^\alpha = \mathbf{a}^\alpha \otimes \mathbf{a}_\alpha$ w.r.t. the corresponding basis vectors.

Up to this point, the two-dimensional mid surface merely provides the kinematic backbone for the approximation of the geometry of the three-dimensional shell continuum. To recover its extension in the thickness direction, we define a three dimensional vector field $\mathbf{d}$ of unit length ($\|\mathbf{d}\| = 1$), called the *shell director* field, in every point of the mid surface:

$$\mathbf{d}\colon \omega \to S^2 \quad \text{with} \quad S^2 := \left\{ \mathbf{x} \in \mathbb{R}^3 \,|\, \|\mathbf{x}\| = 1 \right\}. \tag{9.4}$$

As the mapping $(\chi_1, \chi_2) \mapsto \mathbf{d}(\chi_1, \chi_2)$ takes values in the unit sphere S^2,

$$\mathbf{d} \cdot \mathbf{d} = 1 \quad \Rightarrow \quad \underbrace{\frac{\partial \mathbf{d}}{\chi_\alpha}}_{=:\,\mathbf{d}_{,\alpha}} \cdot \mathbf{d} = 0 \tag{9.5}$$

holds for all $\chi := (\chi_1, \chi_2) \in \omega$. The pair $(\varphi, \mathbf{d})$ of functions defined in (9.1) and (9.4) together with the thickness variable $\zeta \in \left[-\frac{h}{2}, \frac{h}{2}\right]$ determine the three-dimensional configuration of the shell continuum in terms of the vector valued function

$$\phi(\chi, \zeta) := \varphi(\chi) \; + \; \zeta\, \mathbf{d}(\chi), \tag{9.6}$$

which is defined on the three-dimensional parameter domain $\Omega := \omega \times \left[-\frac{h}{2}, \frac{h}{2}\right]$ and provides a parametrization of the position vectors of the material points within the three-dimensional shell configuration $\phi(\Omega) =: \mathscr{S} \subset \mathbb{R}^3$ by curvilinear coordinates. By differentiating the mapping ϕ w.r.t. the variables $(\chi_\alpha, \zeta) \in \Omega$ we obtain the covariant basis vectors

$$\mathbf{g}_\alpha = \frac{\partial}{\partial \chi_\alpha} \phi = \mathbf{a}_\alpha + \zeta\, \mathbf{d}_{,\alpha}, \tag{9.7}$$

$$\mathbf{g}_3 = \frac{\partial}{\partial \zeta} \phi = \mathbf{d} \tag{9.8}$$

associated to the parametrization of the three-dimensional shell continuum. Analogous to the definition of the contravariant base vectors of the mid surface, the system of equations $\mathbf{g}_i \cdot \mathbf{g}^j = \delta_i^j$ uniquely determines the contravariant

counterparts, and the scalar products $g_{ij} := \mathbf{g}_i \cdot \mathbf{g}_j$ and $g^{ij} := \mathbf{g}^i \cdot \mathbf{g}^j$ yield the co- and contravariant components of the metric tensor $\mathsf{Id} = \mathbf{e}_i \otimes \mathbf{e}_i$ of $\mathbb{R}^3$, which may be equivalently represented as: $\mathsf{Id} = g_{ij}\, \mathbf{g}^i \otimes \mathbf{g}^j = g^{ij}\, \mathbf{g}_i \otimes \mathbf{g}_j = \mathbf{g}_j \otimes \mathbf{g}^j = \mathbf{g}^j \otimes \mathbf{g}_j$.

Making use of the parametrization of the three-dimensional shell volume with curvilinear coordinates as described above, we additionally provide the following set of three kinematic quantities, which correspond to the covariant components of two tensor fields and a vector field defined on the shell surface and may be considered as a complete set of *differential invariants*[2] of the latter:

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta, \quad \kappa_{\alpha\beta} := \frac{1}{2}(\mathbf{d}_{,\alpha} \cdot \mathbf{a}_\beta + \mathbf{d}_{,\beta} \cdot \mathbf{a}_\alpha), \quad \gamma_\alpha = \mathbf{a}_\alpha \cdot \mathbf{d}. \tag{9.9}$$

The first term $a_{\alpha\beta}$ corresponds to the already known components of the *metric* tensor on the mid surface. The second term $\kappa_{\alpha\beta}$ can be interpreted as the components of the generalized *curvature* tensor[3] of the shell surface. The last quantity yields the components γ_α of the orthogonal projection $(\mathsf{I} - \mathbf{a}_3 \otimes \mathbf{a}_3)\, \mathbf{d} = (\mathbf{a}^\alpha \otimes \mathbf{a}_\alpha)\, \mathbf{d} = \gamma_\alpha\, \mathbf{a}^\alpha$ of the director onto the tangential plane of the mid surface. Note that $\gamma_\alpha = 0$ holds if $\mathbf{d}$ points along the direction of the normal vector $\mathbf{a}_3$ of the shell surface.

With the help of (9.9), we may rewrite the covariant components of the three-dimensionsal metric tensor of the shell continuum in terms of the following exact and approximating expressions:

$$g_{\alpha\beta} = a_{\alpha\beta} + \zeta \kappa_{\alpha\beta} + \mathscr{O}(\zeta^2), \tag{9.10}$$

$$g_{\alpha 3} = \gamma_\alpha, \tag{9.11}$$

$$g_{33} = 1. \tag{9.12}$$

Furtheron, we will neglect the higher order terms $\mathscr{O}(\zeta^2)$ in (9.10), in accordance with the usual assumptions that the principle curvature radii $R_\alpha = 1/\varkappa_\alpha$ given by the eigenvalues $\varkappa_\alpha$ of the tensor $\kappa_{\alpha\beta}\, \mathbf{a}^\alpha \otimes \mathbf{a}^\beta$ are much larger than h, such that the estimate $0 \leq h/R_\alpha \ll 1$ holds allover the shell surface.

---

[2] From the viewpoint of the *differential geometry of directed surfaces.* in Euclidian space, the corresponding tensor quantities are differential invariants of the directed surface, which uniquely determine its geometry in Euclidian space up to rigig body motions, provided that certain *integrability conditions* are satisfied, generalizing the classical equations of Gauss and Codazzi–Mainardi, required to be fulfilled in the special case $\mathbf{d} \equiv \mathbf{a}_3$, to the case of an independent director field $\mathbf{d}$ (see [9] for details and mathematical proofs).

[3] The *curvature tensor* or second fundamental form of the parametrized surface $\varphi : \omega \to \mathbb{R}^3$ is given by the derivative $\mathrm{d}\mathbf{a}_3$ (also called *Weingarten map*) of the *Gauss map* $\chi \mapsto \mathbf{a}_3(\chi) \in S^2$. The symmetric tensor $\kappa_{\alpha\beta}\, \mathbf{a}^\alpha \otimes \mathbf{a}^\beta = \frac{1}{2}(\mathrm{d}\varphi^T \cdot \mathrm{d}\mathbf{d} + \mathrm{d}\mathbf{d}^T \cdot \mathrm{d}\varphi)$ generalizes the Weingarten map to the case of a directed surface given by $(\varphi, \mathbf{d})$ and reduces to $\mathrm{d}\mathbf{a}_3$ in the special case $\mathbf{d} = \mathbf{a}_3$.

Table 9.1 Notation for the different configurations

	Surface basis	Continuum basis	Kinematic quantities
Reference configuration	$\mathbf{A}_\alpha,\ \mathbf{A}^\beta$	$\mathbf{G}_i,\ \mathbf{G}^j$	$A_{\alpha\beta},\ \mathrm{K}_{\alpha\beta},\ \Gamma_\alpha$
Deformed configuration	$\mathbf{a}_\alpha,\ \mathbf{a}^\beta$	$\mathbf{g}_i,\ \mathbf{g}^j$	$a_{\alpha\beta},\ \kappa_{\alpha\beta},\ \gamma_\alpha$

9.2.2 Strains and Stresses

To measure strains, we define a *reference configuration* $\mathscr{S}_0 \subset \mathbb{R}^3$ of the shell, which we assume to be in a natural (stress free) state, by a parametrization over the parameter domain Ω (following the notational conventions of Sect. 9.2.1) as follows:

$$\phi_0(\chi, \zeta) \ := \ \varphi_0(\chi) \ + \ \zeta\, \mathbf{d}_0(\chi), \quad \mathscr{S}_0 \ := \ \phi_0(\Omega). \tag{9.13}$$

To distinguish the quantities of the different configurations, capital letters are used for those of the stress free shell continuum (see Table 9.1).

We assume that in the reference configuration the director equals the unit normal $\mathbf{A}_3 =: \mathbf{d}_0$ of the mid surface for all $\chi \in \omega$, such that the kinematic quantity $\Gamma_\alpha = 0$ from (9.9) vanishes. This implies that the components (9.11) of the covariant metric are eliminated, as well as their contravariant counter parts $G_{\alpha 3} = G^{\alpha 3} = 0$. The *deformed* configuration $\mathscr{S} \subset \mathbb{R}^3$ of the shell continuum is given by

$$\phi(\chi, \zeta) \ := \ \varphi(\chi) \ + \ \zeta \mathbf{d}(\chi), \quad \mathscr{S} \ := \ \phi(\Omega). \tag{9.14}$$

Different from the reference configuration, we do not assume that the deformed director remains normal to the midsurface, so $\mathbf{d} \neq \mathbf{a}_3$ in general for deformed configurations of the shell. Nevertheless, it always remains normalized due to (9.4).

The functions (9.13) and (9.14) map from the same parameter domain Ω onto $\mathscr{S}$ or $\mathscr{S}_0$. Therefore, the deformation mapping between the reference and the deformed configuration is given by the composition

$$\Phi := \phi \circ \phi_0^{-1} \tag{9.15}$$

Figure 9.1 illustrates the connection of the functions. The derivatives of ϕ with respect to the variable $\mathbf{X} \in \mathscr{S}_0$ determines the deformation gradient $\mathsf{F} := \nabla_{\mathbf{X}} \Phi$. With some tensor algebra [20], this tensor transforms to

$$\mathsf{F} = \mathbf{g}_i \otimes \mathbf{G}^i. \tag{9.16}$$

To detect strains between the two configurations $\mathscr{S}$ and $\mathscr{S}_0$, we choose the *Green-Lagrange tensor* given by

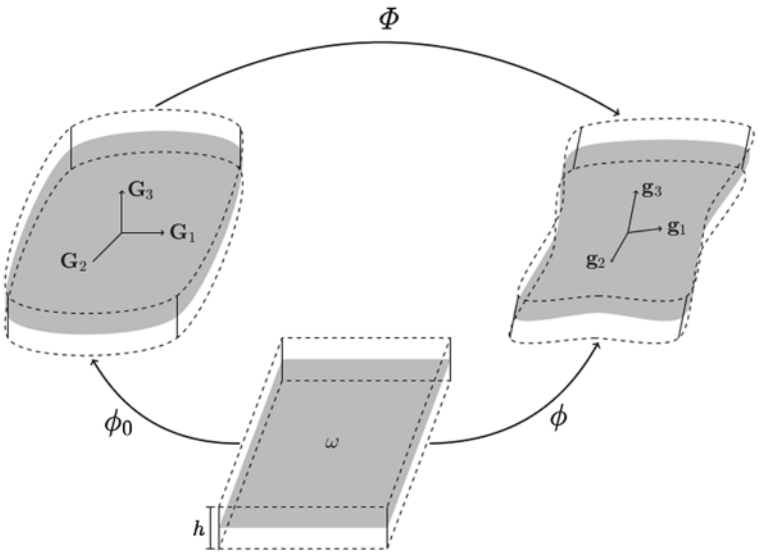


Fig. 9.1 Configurations of the shell continuum

$$\mathsf{E} = \frac{1}{2}\left(\mathsf{F}^T\mathsf{F} - \mathsf{Id}\right), \tag{9.17}$$

$$= \frac{1}{2}\underbrace{\left(g_{ij} - G_{ij}\right)}_{=:E_{ij}}\left(\mathbf{G}^i \otimes \mathbf{G}^j\right). \tag{9.18}$$

as a nonlinear objective strain measure, where the second equality is based on (9.16) and $\mathsf{I} = \mathbf{G}_i \otimes \mathbf{G}^i$. Equation (9.18) indicates explicitely that differences between the two metrics measure the strains, such that a non-vanishing tensor E measures the *change of metric* caused by deformations of the reference configuration.

In the case of a shell, the components of the Green Lagrange tensor read

$$E_{\alpha\beta} = \underbrace{\left(a_{\alpha\beta} - A_{\alpha\beta}\right)}_{=:m_{\alpha\beta}} + \zeta\,\underbrace{\left(\kappa_{\alpha\beta} - \mathrm{K}_{\alpha\beta}\right)}_{=:b_{\alpha\beta}}, \tag{9.19}$$

$$E_{\alpha 3} = \underbrace{\gamma_\alpha - \Gamma_\alpha}_{=:s_\alpha}, \tag{9.20}$$

$$E_{33} = 0, \tag{9.21}$$

neglecting higher order terms $\mathscr{O}(\zeta^2)$ by assuming principle curvature radii much larger than the shell thickness, and local strains that remain small also in deformed configurations. Note that (9.21) implies that there is no strain in normal direction of the reference configuration, due to the unit lengths of $\mathbf{d}_0$ and $\mathbf{d}$.

The 2nd *Piola Kirchoff stress tensor* S is the work conjugate stress measure corresponding to the Green-Lagrange strain E [4]. In accordance with the assumption of locally small strains, one may assume a *linear* constitutive relation of S and E, such that stress and strain are connected according to *generalized Hooke's law* via a symmetric fourth order *elasticity tensor* $\mathbb{C}$ as

$$\mathsf{S} = \mathbb{C} : \mathsf{E}. \tag{9.22}$$

The various assumptions stated above imply a restriction of the model to *small local deformations*. However, large rotations and displacements are still possible, if they only imply small nonlinear strains E_{ij}. Additionally we suppose that the following components of the material elasticity tensor $\mathbb{C} = \mathbb{C}^{ijkl}\mathbf{G}_i \otimes \mathbf{G}_j \otimes \mathbf{G}_k \otimes \mathbf{G}_l$ vanish w.r.t. the covariant basis:

$$\mathbb{C}^{\alpha 333} = \mathbb{C}^{3\alpha 33} = \mathbb{C}^{33\alpha 3} = \mathbb{C}^{333\alpha} = 0, \tag{9.23}$$

$$\mathbb{C}^{\alpha\beta\iota 3} = \mathbb{C}^{\alpha\beta 3\iota} = \mathbb{C}^{\alpha 3\beta\iota} = \mathbb{C}^{3\alpha\beta\iota} = 0. \tag{9.24}$$

Physically linear constitutive laws for a geometrically exact shell are discussed in more detail in Sect. 9.3. With (9.23) and (9.24) the components of the 2nd Piola Kirchhoff stress tensor w.r.t. the covariant basis $\mathbf{G}_i \otimes \mathbf{G}_j$ are given by

$$S^{\alpha\beta} = \mathbb{C}^{\alpha\beta\iota\pi}E_{\iota\pi} + \mathbb{C}^{\alpha\beta 33}E_{33}, \tag{9.25}$$

$$S^{\alpha 3} = \mathbb{C}^{\alpha 3\iota 3}E_{\iota 3}, \tag{9.26}$$

$$S^{33} = \mathbb{C}^{33\iota\pi}E_{\iota\pi} + \mathbb{C}^{3333}E_{33}. \tag{9.27}$$

In addition to plain strain ($E_{33} = 0$), we assume that there is likewise no stress in the normal direction of the reference configuration $S^{33} = 0$ (plain stress). Therefore (9.27) may be solved w.r.t. E_{33}, which is then substituted back into (9.25). With the definitions

$$\mathbb{H}^{\alpha\beta\iota\pi} := \mathbb{C}^{\alpha\beta\iota\pi} + \frac{\mathbb{C}^{\alpha\beta 33}\mathbb{C}^{33\iota\pi}}{\mathbb{C}^{3333}}, \tag{9.28}$$

$$\hat{\mathbb{H}}^{\alpha\beta} := \mathbb{C}^{\alpha 3\beta 3}, \tag{9.29}$$

the stress strain relations (9.25)–(9.27) for the shell continuum transform to

$$S^{\alpha\beta} = \mathbb{H}^{\alpha\beta\iota\pi}E_{\iota\pi}, \tag{9.30}$$

$$S^{\alpha 3} = \hat{\mathbb{H}}^{\alpha\beta}E_{\beta 3}. \tag{9.31}$$

Altogether membrane and bending strains lead to in-plane stresses, and shear strains are the cause of transverse stresses within the shell.

9.2.3 Weak Formulation of Balance of Momentum

Following [21], the weak form of balance of linear momentum for three dimensional continua is given by

$$G(\phi, \delta\phi) := \int_{\mathscr{S}_0} \mathsf{S} : \delta\mathsf{E}\, \mathrm{d}\mathbf{X} + \int_{\mathscr{S}_0} \rho_0 \ddot{\Phi} \cdot \delta\Phi\, \mathrm{d}\mathbf{X} - \int_{\mathscr{S}_0} \mathbf{B} \cdot \delta\Phi\, \mathrm{d}\mathbf{X} = 0. \tag{9.32}$$

The vector $\mathbf{B}$ represents the external loads, and ρ_0 denotes the mass density of the shell with respect to the reference configuration. The variation of the deformation map is given by $\delta\Phi = \delta\phi \circ \phi_0^{-1}$ with $\delta\phi = \delta\varphi + \zeta\delta\mathbf{d}$.

The variation of the director $\delta\mathbf{d}$ maps onto $T_{\mathbf{d}}S^2$, i.e.: the tangential space of S^2 in $\mathbf{d}$, as $\mathbf{d} \cdot \delta\mathbf{d} = 0$ holds for all $\delta\mathbf{d} \in T_{\mathbf{d}}S^2$. The variation of the kinematic quantities and strain measures can be calculated in a straightforward manner:

$$\delta a_{\alpha\beta} = \delta\mathbf{a}_\alpha \cdot \mathbf{a}_\beta + \mathbf{a}_\alpha \cdot \delta\mathbf{a}_\beta, \tag{9.33}$$

$$2\delta\kappa_{\alpha\beta} = \delta\mathbf{a}_\alpha \cdot \mathbf{d}_{,\beta} + \mathbf{a}_\alpha \cdot \delta\mathbf{d}_{,\beta} + \delta\mathbf{a}_\beta \cdot \mathbf{d}_{,\alpha} + \mathbf{a}_\beta \cdot \delta\mathbf{d}_{,\alpha}, \tag{9.34}$$

$$\delta\gamma_\alpha = \delta\mathbf{a}_\alpha \cdot \mathbf{d} + \mathbf{a}_\alpha \cdot \delta\mathbf{d}. \tag{9.35}$$

The part of (9.32) dealing with the internal forces may be rewritten with the help of integration by substitution as

$$G_{\mathrm{int}}(\phi, \delta\phi) := \int_{\mathscr{S}_0} \mathsf{S} : \delta\mathsf{E}\, \mathrm{d}\mathbf{X} = \int_{\Omega} \mathsf{S} : \delta\mathsf{E}\, \det(G_{ij})\, \mathrm{d}\zeta\, \mathrm{d}\chi. \tag{9.36}$$

Now we want to reduce the problem by integrating analytically over the variable ζ. The integrand of (9.36) decomposes to

$$\mathsf{S} : \delta\mathsf{E} = a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi} + \gamma_\alpha \hat{\mathbb{H}}^{\alpha\beta}\delta\gamma \tag{9.37}$$

$$+ \zeta \left(a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi} + \kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi} \right) \tag{9.38}$$

$$+ \zeta^2 \kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi}. \tag{9.39}$$

Additionally $\mathbb{H}$ and $\hat{\mathbb{H}}$ depend on ζ via G^{ij} (see Sect. 9.3). Next we approximate the continuum metric by the midsurface metric $G^{ij} \approx A^{ij}$, which are equivalent in the limit $h \to 0$. With this approach the only dependencies on ζ are those explicitly written in (9.37)–(9.39), and we are able to perform the integration over ζ in closed form. Due to the symmetry of the integration domain $[-h/2, h/2]$, it holds that

$$\int_{-h/2}^{h/2} \zeta \left(a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi} + \kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi} \right) \mathrm{d}\zeta = 0, \tag{9.40}$$

which decouples membrane and bending deformations of the shell surface. Integrating (9.37) and (9.39) yields

$$\int_{-h/2}^{h/2} a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi}\,\mathrm{d}\zeta = h\, a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi}, \tag{9.41}$$

$$\int_{-h/2}^{h/2} \gamma_\alpha \hat{\mathbb{H}}^{\alpha\beta}\delta\gamma\mathrm{d}\zeta = h\, \gamma_\alpha \hat{\mathbb{H}}^{\alpha\beta}\delta\gamma \tag{9.42}$$

$$\int_{-h/2}^{h/2} \zeta^2\kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi}\,\mathrm{d}\zeta = \frac{h^3}{12}\,\kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi}. \tag{9.43}$$

Altogether, Eq. (9.36) reduces to a two dimensional integral

$$G_{\mathrm{int}}(\phi,\delta\phi) = h\int_\omega \left(a_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta a_{\iota\pi} + \gamma_\alpha\hat{\mathbb{H}}^{\alpha\beta}\delta\gamma\right)\det(A_{\alpha\beta})\mathrm{d}\chi \tag{9.44}$$

$$+\frac{h^3}{12}\int_\omega \kappa_{\alpha\beta}\mathbb{H}^{\alpha\beta\iota\pi}\delta\kappa_{\iota\pi}\det(A_{\alpha\beta})\mathrm{d}\chi \tag{9.45}$$

over the shell surface. The two other summands of (9.32) may be integrated over ζ in a similar way. The dynamical part of (9.32) reduces to the surface integral

$$G_{\mathrm{dyn}}(\ddot{\phi},\delta\phi) := \int_{\mathscr{S}_0}\rho_0\ddot{\Phi}\cdot\delta\Phi\;\mathrm{d}\mathbf{X} \tag{9.46}$$

$$= \int_\omega \left(\mathsf{A}_{\rho_0}\ddot{\phi}\cdot\delta\phi + \mathsf{I}_{\rho_0}\ddot{\mathbf{d}}\cdot\delta\mathbf{d}\right)\det(A_{\alpha\beta})\mathrm{d}\chi, \tag{9.47}$$

with the nominal surface mass density $\mathsf{A}_{\rho_0}(\chi) := \int\rho_0(\chi,\zeta)\,\mathrm{d}\zeta$, and the corresponding rotational inertia $\mathsf{I}_{\rho_0}(\chi) = \int\zeta^2\rho_0(\chi,\zeta)\,\mathrm{d}\zeta$. Assuming the density ρ_0 of the reference shell continuum to be homogeneous along the normal direction of the midsurface (i.e.: independent of ζ), the inertial parameters simplify to the well known expressions $\mathsf{A}_{\rho_0} = h\rho_0$ and $\mathsf{I}_{\rho_0} = \frac{h^3}{12}\rho_0$, both in general still depending on the surface parameters χ, or constants if ρ_0 is constant. The part of (9.32) dealing with the external forces may be integrated in the same way, therefore it is skipped here.

9.3 Linear Elastic Materials for Geometrically Exact Shells

The main content of this section is the discussion of elastic constitutive equations, as introduced in (9.22), for geometrically exact shells. We focus on transferring constitutive laws from the three dimensional theory [12] to the shell continuum and converting it to a two dimensional material law as in (9.28) and (9.29). As mentioned in Sect. 9.2.2, we restrict our considerations to a linear relation between stresses and strains. In the following we present two well known classes of linear materials. First we deal with the standard *isotropic* Saint-Vernant-Kirchhoff material, to illustrate the procedure. Additionally we present a certain type of *orthotropic* material law, which later enables us to deal with the material anisotropy encountered within the reinforced carcass and belt structures of real tires.

9.3.1 Linear Elastic Isotropic Materials

The equivalence of all directions is the basic property of *isotropic* constitutive behavior, such that material properties may be expressed independent of any particular material reference frame.

For the elasticity tensor $\mathbb{C}_{\text{iso}}$ corresponding to Hooke's law for isotropic materials this implies an additive decomposition into two parts:

- One part is proportional to the tensor $\mathsf{Id} \otimes \mathsf{Id}$, which acts on arbitrary second order tensors A according to $(\mathsf{Id} \otimes \mathsf{Id})\colon \mathsf{A} = \mathrm{Tr}(\mathsf{A})\,\mathsf{Id}$.
- The other part is proportional to the sum $\mathbb{I} + \mathbb{T}$ of the fourth order identity tensor $\mathbb{I}$, which maps second order tensors A identically to themselves (i.e.: $\mathbb{I}\colon \mathsf{A} = \mathsf{A}$), and the fourth order transposition tensor $\mathbb{T}$, mapping second order tensors to their transpose (i.e.: $\mathbb{T}\colon \mathsf{A} = \mathsf{A}^T$), such that the symmetric part of A is obtained as $\frac{1}{2}(\mathsf{A} + \mathsf{A}^T) = \frac{1}{2}(\mathbb{I} + \mathbb{T})\colon \mathsf{A}$.
- The respective proportionality parameters L_1 and L_2, called *Lamé parameters*[4] correspond to the *two independent elastic moduli* governing the isotropic material behaviour for small strains.

Expressing the Lamé parameters by *Young's modulus E* and *Poisson's ratio* ν, the isotropic elasticity tensor of the *Saint–Venant–Kirchhoff* material may be written as

$$\mathbb{C}_{\text{iso}} = \underbrace{\frac{E\nu}{(1+\nu)(1-2\nu)}}_{=L_1} (\mathsf{Id} \otimes \mathsf{Id}) + \underbrace{\frac{E}{2(1+\nu)}}_{=L_2} (\mathbb{I} + \mathbb{T}), \tag{9.48}$$

[4] Usually the Lamé parameters are denoted as λ and $\mu \equiv G$. We follow the notation used in Sect. 4.2.1 of [6] to avoid notational overlap with Greek indices and improve notational similarity with the orthotropic case with parameters L_α^{ij} appearing in the material tensor.

with the corresponding stress-strain relation $\mathsf{S} = \mathbb{C}_{\text{iso}} : \mathsf{E} = L_1 \operatorname{Tr}(\mathsf{E})\,\mathsf{Id} + 2L_2\,\mathsf{E}$, and the components of $\mathbb{C}_{\text{iso}}$ w.r.t. the contravariant basis given by:

$$\mathbb{C}_{\text{iso}}^{ijkl} := \left(\mathbf{G}_i \otimes \mathbf{G}_j\right) : \mathbb{C}_{\text{iso}} : \left(\mathbf{G}_k \otimes \mathbf{G}_l\right) \tag{9.49}$$

$$= L_1\, G^{ij} G^{kl} \;+\; L_2 \left(G^{ik} G^{jl} + G^{il} G^{kj}\right). \tag{9.50}$$

Obviously (9.23) and (9.24) hold because of $G^{\alpha 3} = 0$. Hence, the isotropic material tensor $\mathbb{C}_{\text{iso}}$ transforms under the plain stress assumption to the shell material tensors $\mathbb{H}_{\text{iso}}^{\alpha\beta\iota\pi}$ and $\hat{\mathbb{H}}_{\text{iso}}^{\alpha\beta}$ (see Sect. 9.2.2 and the Appendix for explicit formulas).

9.3.2 Linear Elastic Orthotropic Materials

In contrast to the isotropic case with equal material behaviour in all spatial directions, an *orthotropic* material posesses only *three orthogonal planes* of symmetry, corresponding to unit vectors $\mathbf{E}_i$ along the normal directions of these planes.

The triple $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ determines a specific orthonormal basis, calles the *material reference frame*. To each of these material directions $\mathbf{E}_i$ belongs one Young's modulus E_i, and to each symmetry plane there is a pair of Poisson ratios ν_{ij} and ν_{ji} as well as an associated shear modulus G_{ij}. Together these *nine independent elastic parameters* determine the coefficients L_α^{ij} of the orthotropic material tensor w.r.t. the material reference frame according to the following relations (see [5, 11]):

$$\begin{aligned}
L_1^{11} &= -E_1 \frac{1 - E_1\nu_{23}\nu_{32}}{\Delta} & L_1^{22} &= -E_2 \frac{1 - E_2\nu_{13}\nu_{31}}{\Delta} & L_1^{33} &= -E_3 \frac{1 - E_3\nu_{12}\nu_{21}}{\Delta} \\
L_1^{12} &= -E_1 \frac{\nu_{12} + \nu_{13}\nu_{32}}{\Delta} & L_1^{13} &= -E_1 \frac{\nu_{13} + \nu_{12}\nu_{23}}{\Delta} & L_1^{23} &= -E_2 \frac{\nu_{23} + \nu_{13}\nu_{21}}{\Delta} \\
L_1^{21} &= -E_2 \frac{\nu_{21} + \nu_{23}\nu_{31}}{\Delta} & L_1^{31} &= -E_3 \frac{\nu_{31} + \nu_{21}\nu_{32}}{\Delta} & L_1^{32} &= -E_3 \frac{\nu_{32} + \nu_{13}\nu_{31}}{\Delta}
\end{aligned} \tag{9.51}$$

with

$$\Delta \;=\; \nu_{12}\nu_{21} + \nu_{13}\nu_{31} + \nu_{23}\nu_{32} + \nu_{12}\nu_{23}\nu_{31} + \nu_{13}\nu_{32}\nu_{21} \tag{9.52}$$

and

$$L_2^{12} = L_2^{21} = G_{12}, \quad L_2^{13} = L_2^{31} = G_{13}, \quad L_2^{23} = L_2^{31} = G_{23}. \tag{9.53}$$

Due to the symmetry of the material, the relations $G_{ij} = G_{ji}$ and $\nu_{ji} = (E_i/E_j)\nu_{ij}$ hold (see [12]). Consequently the coefficients $L_1^{ij} = L_1^{ji}$ and $L_2^{ij} = L_2^{ji}$ are likewise symmetric. Altogether the orthotropic material tensor may be written as

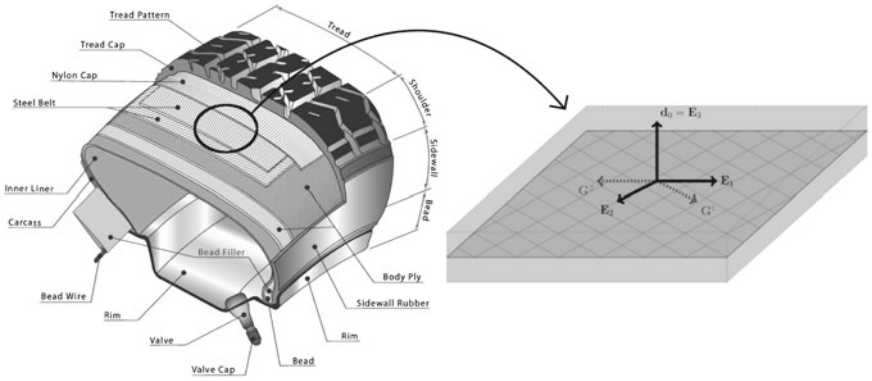


Fig. 9.2 Special orthotropy of a tire with condensed reinforcements

$$\mathbb{C}_{\text{orth}} = \sum_{i=1}^{3}\sum_{j=1}^{3} L_1^{ij}(\mathbf{E}_i \otimes \mathbf{E}_i \otimes \mathbf{E}_j \otimes \mathbf{E}_j) \tag{9.54}$$

$$+ \sum_{i=1}^{3}\sum_{j=1}^{3} L_2^{ij}(\mathbf{E}_i \otimes \mathbf{E}_j \otimes \mathbf{E}_i \otimes \mathbf{E}_j + \mathbf{E}_i \otimes \mathbf{E}_j \otimes \mathbf{E}_j \otimes \mathbf{E}_i) \tag{9.55}$$

w.r.t. the orthonormal material basis. As in the isotropic case (9.49), the coefficients of the tensor w.r.t. the contravariant basis may be computed from

$$\mathbb{C}_{\text{orth}}^{ijkl} := \left(\mathbf{G}_i \otimes \mathbf{G}_j\right) : \mathbb{C}_{\text{orth}} : (\mathbf{G}_k \otimes \mathbf{G}_l)\,. \tag{9.56}$$

To transfer the material law to the shell, Eqs. (9.23) and (9.24) must hold. This is ensured by the assumption that one principal direction of the orthotropic material law e.g. $\mathbf{E}_3$ is parallel to the director $\mathbf{d} = \mathbf{G}^3$ in every point of the reference configuration, which holds as a first approximation of the structure of the tire, if one considers the material properties of the various plys to be condensed to the mid surface. Additionaly we assume that the reinforcements are mirror-symmetric in longitudinal and lateral direction of the tire, as illustrated in Fig. 9.2, which is also typically the case in real passenger car tires.

With these assumptions, the following scalar products vanish:

$$\mathbf{E}_\alpha \cdot \mathbf{G}^3 = 0 \quad \mathbf{E}_3 \cdot \mathbf{G}^\alpha = 0. \tag{9.57}$$

This in turn implies that also $\mathbb{C}_{\text{orth}}^{\alpha 333} = 0$ and $\mathbb{C}_{\text{orth}}^{\alpha\beta\iota 3} = 0$ hold. To evaluate the remaining coefficients, one has to compute the scalar products

$$B_\iota^\alpha = \mathbf{E}_\iota \cdot \mathbf{G}^\alpha. \tag{9.58}$$

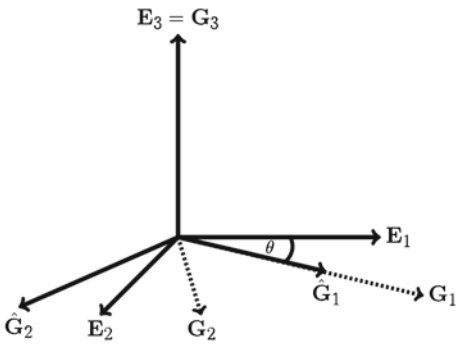


Fig. 9.3 Relation of the basis

To simplify this task, we interrelate the material $\mathbf{E}_\alpha$ and covariant $\mathbf{G}_\alpha$ shell basis as follows: Both pairs of vectors span the same two dimensional plane, because of the assumption $\mathbf{G}_3 = \mathbf{G}^3 = \mathbf{E}_3$. In contrast to $\mathbf{E}_\alpha$, the tangential vectors $\mathbf{G}_\alpha$ do not have to be perpendicular or normalized. Therefore we introduce the pair

$$\hat{\mathbf{G}}_1 = \frac{1}{\sqrt{G_{11}}}\mathbf{G}_1 \quad \hat{\mathbf{G}}_2 = \frac{1}{\sqrt{G^{22}}}\mathbf{G}^2, \tag{9.59}$$

of vectors, which together with $\mathbf{G}_3$ form an orthonormal basis. As shown in Fig. 9.3, a rotation around $\mathbf{G_3}$ by the angle θ transforms the material basis to $\hat{\mathbf{G}}_\alpha$. Therefore the principal material directions are given by

$$\mathbf{E}_1 = \cos(\theta)\hat{\mathbf{G}}_1 + \sin(\theta)\hat{\mathbf{G}}_2 \quad \mathbf{E}_2 = -\sin(\theta)\hat{\mathbf{G}}_1 + \cos(\theta)\hat{\mathbf{G}}_2. \tag{9.60}$$

Hence, the basis $\mathbf{E}_i$ is completely defined by the angle θ, which determines the relation[5] to the covariant basis of the reference shell continuum. Finally Eq. (9.58) transforms to

$$B_1^1 = \frac{\cos(\theta)}{\sqrt{G_{11}}} + \frac{\sin(\theta)G^{12}}{\sqrt{G^{22}}}, \quad B_1^2 = \sin(\theta)\sqrt{G^{22}} \tag{9.61}$$

$$B_2^1 = \frac{\cos(\theta)G^{12}}{\sqrt{G^{22}}} - \frac{\sin(\theta)}{\sqrt{G_{11}}}, \quad B_2^2 = \cos(\theta)\sqrt{G^{22}} \tag{9.62}$$

Explicit expressions for the corresponding coefficients $\mathbb{H}^{\alpha\beta\iota\pi}_{\text{orth}}$ and $\hat{\mathbb{H}}^{\alpha\beta}_{\text{orth}}$ are given in the Appendix.

[5] Without loss of generality, we define this angle by the relation $\alpha = \cos^{-1}(\hat{\mathbf{G}}_1 \cdot \mathbf{E}_1)$, which is always well defined due to the symmetry of the material.

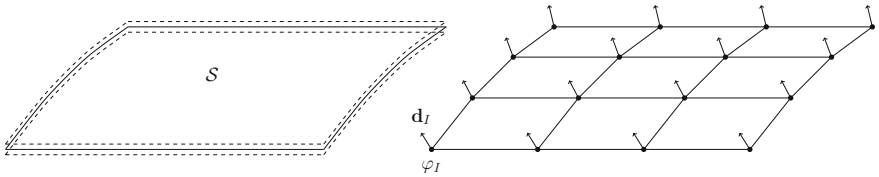


Fig. 9.4 Shell discretized by quadrilateral finite elements

9.4 The Discrete Shell Model

To solve the weak balance law developed in Sect. 9.2.3, we discretize the problem in space and time. For the spatial part finite elements are used, as discussed in the first section. In the second one we present a suitable time integration scheme. In the last section we show how the size of the arising system of nonlinear equations can be reduced.

9.4.1 Spatial Discretization

For the spatial discretization we follow the work from Betsch and Sänger [2], where bilinear isoparametric elements are used to discretize the shell. This approach yields a rotation free formulation, with the peculiarity that we obtain an additional constraint equation for the shell director field. For simplicity we first focus on the static problem, which is given by

$$G_{\text{static}}(\phi, \delta\phi) \;:=\; G_{\text{int}}(\phi, \delta\phi) - G_{\text{ext}}(\delta\phi) \;=\; 0 \quad \forall \delta\phi. \tag{9.63}$$

The midsurface and the director are discretized by the same piecewise bilinear Lagrangian shape function N^I with associated nodal values $\mathbf{q}_I = (\varphi_I, \mathbf{d}_I) \in \mathbb{R}^6$

$$\varphi^h(\chi) \;=\; \sum_I^n \varphi_I \, N^I(\chi) \quad \mathbf{d}^h(\chi) \;=\; \sum_I^n \mathbf{d}_I \, N^I(\chi), \tag{9.64}$$

where n is the number of nodes of the finite element mesh. Figure 9.4 illustrates this discretization scheme.

All other functions and quantities are discretized in the same way, see Table 9.2. So the configuration of the shell is fully described by the state vector

$$\mathbf{q} \;=\; (\mathbf{q}_1 \ldots \mathbf{q}_n)^T \in \mathbb{R}^{6n}. \tag{9.65}$$

Table 9.2 Notation for the different configurations

	Discretization	Discrete variation
$\mathbf{a}_\alpha$	$\mathbf{a}^h_\alpha = \phi_I N^I_{,\alpha}$	$\delta\mathbf{a}^h_\alpha = \delta\phi_I N^I_{,\alpha}$
$a_{\alpha\beta}$	$a^h_{\alpha\beta} = \phi_I \cdot \phi_J N^I_{,\alpha} N^J_{,\beta}$	$\delta a^h_{\alpha\beta} = (\phi_I \cdot \delta\phi_J + \delta\phi_I \cdot \phi_J) N^I_{,\alpha} N^J_{,\beta}$
$\kappa_{\alpha\beta}$	$\kappa^h_{\alpha\beta} = (\phi_I \cdot \mathbf{d}_J + \mathbf{d}_I \cdot \phi_J) N^I_{,\alpha} N^J_{,\beta}$	$\delta\kappa^h_{\alpha\beta} = (\phi_I \cdot \delta\mathbf{d}_J + \delta\phi_I \cdot \mathbf{d}_J + \mathbf{d}_I \cdot \delta\phi_J + \delta\mathbf{d}_I \cdot \phi_J) N^I_{,\alpha} N^J_{,\beta}$
γ_α	$\gamma^h_\alpha = \phi_I \cdot \mathbf{d}_J N^I_{,\alpha} N^J$	$\delta\gamma^h_\alpha = (\phi_I \cdot \delta\mathbf{d}_J + \delta\phi_I \cdot \mathbf{d}_J) N^I_{,\alpha} N^J$

By inserting (9.64) into (9.63), we get a discrete version of the static problem

$$G_{\text{static}}(\phi_h, \delta\phi^h) = \left(\mathbf{R}_J(\mathbf{q}) - \mathbf{F}^h_J\right) \cdot \delta\mathbf{q}_J \;=\; 0. \tag{9.66}$$

The nonlinear part of this equation arises from the internal part $G_{\text{int}}(\phi^h, \delta\phi^h)$ of the static equation

$$\mathbf{R}_J(\mathbf{q}) \cdot \delta\mathbf{q}_J \;=\; \begin{pmatrix} \int\limits_\omega (2S^{\alpha\beta}_h \mathbf{a}^h_\alpha + 2S^{\alpha\beta}_h \mathbf{d}^h_{,\alpha} + S^{3\beta}\mathbf{d}^h) N^J_\beta \,\mathrm{d}\chi \\ \int\limits_\omega (2S^{\alpha\beta}_h \mathbf{a}^h_\alpha N^J_{,\beta} + S^{3\alpha}\mathbf{a}^h_\alpha) N^J \,\mathrm{d}\chi \end{pmatrix} \cdot \begin{pmatrix} \delta\phi_J \\ \delta\mathbf{d}_J \end{pmatrix} \tag{9.67}$$

Integrating the external force over the shell continuum leads to the discrete external force $\mathbf{F}^h_J = \int \mathbf{B} N^J \,\mathrm{d}\chi$. Because (9.66) holds for all $\delta\mathbf{q}_J \in R^6$, its solution is equivalent to solving the following nonlinear system of equations:

$$\mathbf{R}(\mathbf{q}) - \mathbf{F} = 0, \tag{9.68}$$

with

$$\mathbf{R}(q) = (\mathbf{R}_1(\mathbf{q}) \ldots \mathbf{R}_n(\mathbf{q}))^T \tag{9.69}$$

$$\mathbf{F} = (\mathbf{F}_1 \ldots \mathbf{F}_n)^T \tag{9.70}$$

In contrast to the continuous case discussed in Sect. 9.2, the discrete directors $\mathbf{d}^h$ have been considered as arbitrary three-dimensional vectors up to this point. In continuous configurations of the shell, they are pointwise restricted to positions located on the unit sphere. In the discrete case, we merely enforce the *unit length* condition at the *nodes* of the finite element mesh. This imposed a set of algebraic constraints on the discrete configurations of the shell, which we incorporate into the governing equations with the help of corresponding Lagrangian multipliers added within the variational formulation. This leads to n additional constraint equations

$$\Lambda_J(\mathbf{q}) \;:=\; \frac{1}{2}(\mathbf{d}_J \cdot \mathbf{d}_J - 1) \;=\; 0 \quad J = 1, \ldots, n. \tag{9.71}$$

The product of the matrix

$$G^T := \nabla_{\mathbf{q}} \Lambda \;=\; \left(\nabla_{\mathbf{q}} \Lambda_1, \ldots, \nabla_{\mathbf{q}} \Lambda_n\right) \in \mathbb{R}^{6n \times n}, \tag{9.72}$$

$$\nabla_{\mathbf{q}} \Lambda_I \;=\; (0 \ldots 0, \mathbf{d}_I^T, 0 \ldots 0)^T \tag{9.73}$$

and the Lagrangian multipliers $\lambda \in \mathbb{R}^n$ contribute an additional constraint force to the system (9.68). Finally, the nonlinear problem

$$\mathbf{R}(\mathbf{q}) - \mathbf{F} + G^T \lambda = 0 \tag{9.74}$$

$$\Lambda(\mathbf{q}) = 0 \tag{9.75}$$

must be solved w.r.t. $\mathbf{q}$ and λ, to obtain a solution of the discrete static problem (9.66). Resolving this problem is done by Newton's Method. The stiffness matrix K is computed by taking the Fréchet derivative of (9.32) and inserting the discretized linearization $\Delta\phi^h$, which is equivalent to the discrete variation $\delta\phi^h$. We refer to [17, 21] for details of this procedure.

Remark 1 To avoid locking, we us the approach from Bathe and Dvorkin [8] for the discrete shear strains γ_α^h and their variations $\delta\gamma_\alpha^h$. Therein the shear strains are evaluated at the edge midpoints along the edge directions and interpolated linear in their corresponding directions.

Remark 2 In case of an orthotropic material law, the θ could vary inside the element, even if the orthonormal frame is constant, because of the change of the discrete tangential vector inside the quadrilateral element. In case of a tire we circumvent the problem by defining the angle in relation to the two parallel edges arising from the revolution of the cross section. Therefore θ is constant and the material is homogeneous at least inside our elements.

9.4.2 Time Integration

In contrast to the static problem (9.63), in a transient simulation the dynamical contribution $G_{\text{dyn}}(\ddot{\phi}, \delta\phi)$ to the weak balance law (9.32) must be considered, too. By this term a second order derivative of the deformation map is involved. Hence, we get a second order ordinary differential equation (*ODE*) w.r.t. the time domain. We use the *vertical method of lines*, meaning that we first perform a discretization in space (see Sect. 9.4.1), and after that the discrete system is integrated in time. Explicitly, the coefficients of the discrete mappings (9.64) become time dependent:

$$\varphi^h(\chi, t) \;=\; \sum_I^n \varphi_I(t)\, N^I(\chi) \quad \mathbf{d}^h(\chi, t) \;=\; \sum_I^n \mathbf{d}_I(t)\, N^I(\chi). \tag{9.76}$$

Therefore, the state vector of the discrete system as well as the Lagrangian multiplier $\lambda(t)$ vary in time $\mathbf{q}(t) \in \mathbb{R}^{6n}$, too. Inserting the time dependent discrete deformation mapping into (9.32), we end up with a differential algebraic equation (*DAE*)

$$\begin{aligned} \mathsf{M}\ddot{\mathbf{q}} + \mathsf{G}^T(\mathbf{q})\lambda + \mathbf{R}(\mathbf{q}) - \mathbf{F}^h &= 0 \\ \Lambda(\mathbf{q}) &= 0. \end{aligned} \tag{9.77}$$

The matrix $\mathsf{M} \in \mathbb{R}^{6n\times 6n}$ arises from the spatial discretization of the dynamic part (9.47), consisting of the block matrices

$$\mathsf{M}_{IJ} = \begin{pmatrix} \int\limits_{\omega} N_I \cdot N_J \mathsf{A}_{\rho_0} \mathrm{d}\chi & 0_{3\times 3} \\ 0_{3\times 3} & \int\limits_{\omega} N_I \cdot N_J \mathsf{I}_{\rho_0} \mathrm{d}\chi \end{pmatrix}. \tag{9.78}$$

By rewriting (9.77) with the help of the generalized nodal velocity $\mathbf{v} = \dot{\mathbf{q}}$ as auxiliary variable, only first order time derivatives appear in the equivalent system of equations

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v} \\ \mathsf{M}\dot{\mathbf{v}} &= -\mathsf{G}^T(\mathbf{q})\lambda - \mathbf{R}(\mathbf{q}) + \mathbf{F}^h \\ 0 &= \Lambda(\mathbf{q}). \end{aligned} \tag{9.79}$$

We discretize (9.79) with the help of an implicit time integration scheme, because of the stiffness of the problem. In contrast to [2], where the focus lies on energy conservation, we use a backward differential formula (*BDF*). This family of time integration schemes are multi step methods, as also results from previous time steps are used. The BDF method is A-stable up to order two and $A(\alpha)$-stable up to order six [19]. To describe the method, we restrict ourselves to the representative time interval $[t_k, t_{k+1}]$, where $\tau := t_{k+1} - t_k$ denotes the constant[6] time step. Furthermore, $(\mathbf{q}^k, \mathbf{v}^k, \lambda^k)$ denote the state vector, the velocities and the Lagrangian multiplier at time t_k. To perform the time step $t_k \to t_{k+1}$ the following system of equations defines the states at t_{k+1}

$$\sum_{i=0}^{m} \alpha_i \mathbf{q}^{k+1-i} = \tau \mathbf{v}_{k+1}, \tag{9.80}$$

$$\mathsf{M} \sum_{i=0}^{m} \alpha_i \mathbf{v}^{k+1-i} = \tau \left(-\mathbf{R}\left(\mathbf{q}^{k+1}\right) - \mathsf{G}^T\left(\mathbf{q}^{k+1}\right) \lambda^{k+1} + \mathbf{F}^h \right), \tag{9.81}$$

$$0 = \Lambda(\mathbf{q}^{k+1}), \tag{9.82}$$

[6] A variable step size is also possible in BDF. For a simpler notation, we only show the constant version.

where m is the order of the BDF method and α_i its coefficients, see [19] for more details. Equation (9.80) relates $\mathbf{v}^{k+1}$ explicitly to $\mathbf{q}^{k+1}$. Therefore the left hand side of (9.81) gets

$$\mathsf{M} \sum_{i=0}^{m} \frac{\alpha_i}{\tau} \mathbf{v}^{k+1-i} = \mathsf{M} \left(\frac{\alpha_0}{\tau}\right)^2 \mathbf{q}^{k+1} + \underbrace{\mathsf{M} \sum_{i=0}^{m-1} \alpha_{i+1} \left(\mathbf{v}^{k-i} + \frac{\alpha_0}{\tau} \mathbf{q}^{k-i}\right)}_{=:\mathbf{R}_{dyn}^k}. \tag{9.83}$$

The term $\mathbf{R}_{dyn}^k$ only depends on the already calculated states and velocities of the past time steps t_{k-i}. With (9.83) the system (9.80)–(9.82) reduces to

$$\mathsf{M} \left(\frac{\alpha_0}{\tau}\right)^2 \mathbf{q}^{k+1} + \mathbf{R}_{dyn}^k = -\mathbf{R}(\mathbf{q}^{k+1}) - \mathsf{G}^T(\mathbf{q}^{k+1})\lambda^{k+1} + \mathbf{F}^h, \tag{9.84}$$

$$0 = \Lambda(\mathbf{q}^{k+1}). \tag{9.85}$$

These nonlinear relations are solved with respect to the states $\mathbf{q}^{k+1}$ and the Lagrangian multiplier λ^{k+1} at the next time step t_{k+1}. Afterwards, the velocities $\mathbf{v}^{k+1}$ are calculate by (9.80). By introducing the nonlinear function

$$\mathbf{R}_{dyn}(\mathbf{q}^{k+1}) := \mathbf{R}(\mathbf{q}^{k+1}) + \left(\frac{\alpha_0}{\tau}\right)^2 \mathsf{M}\, \mathbf{q}^{k+1}$$

and the vector $\mathbf{F}_{dyn}^h := \mathbf{F}^h - \mathbf{R}_{dyn}^k$, which is constant within the time step, the system of equations

$$\mathbf{R}_{dyn}(\mathbf{q}^{k+1}) + \mathsf{G}^T(\mathbf{q}^{k+1})\lambda^{k+1} - \mathbf{F}_{dyn}^h = 0, \tag{9.86}$$

$$\Lambda(\mathbf{q}^{k+1}) = 0, \tag{9.87}$$

which has to be solved to determine the states at t_{k+1}, has precisely the same structure as Eqs. (9.74)–(9.75) to be solved in the static case. This implies that, in every time step we have to solve a perturbed static problem. This is done as in the static case with the help of Newton's method, using the states of the previous time step $\mathbf{q}^k$ as start values for the iteration. Note that the constant vectors of (9.86)–(9.87) are merely treated as constants within a time step. They depend on the states at past time steps (9.83) and could also depend on time directly.

9.4.3 Size Reduction

In the static case (9.74)–(9.75) as well as inside the time step (9.86)–(9.87), the nonlinear system of equations to be solved have the same structure. Therefore we examine the size reduction in the static case, keeping in mind that the same is valid for the dynamical system inside the time step. We achieve size reduction by using a *discrete nullspace matrix* combined with a *local (re)parametrization* of the nodal values, as suggested in the works [2, 3, 13].

First we get rid of the constraint forces together with the Lagrangian multipliers, with the help of a discrete null space matrix $\mathsf{P}(\mathbf{q}) \in \mathbb{R}^{6n\times 5n}$. In our case this matrix reads

$$\mathsf{P}_I = \begin{pmatrix} \mathsf{Id}_{3\times 3} & 0_{3\times 1} & 0_{3\times 1} \\ 0_{3\times 3} & \hat{\mathbf{d}}_I^1 & \hat{\mathbf{d}}_I^2 \end{pmatrix} \in \mathbb{R}^{6\times 5} \quad \mathsf{P} = \begin{pmatrix} \mathsf{P}_1 & 0_{6\times 5} & \cdots & 0_{6\times 5} \\ 0_{6\times 5} & \mathsf{P}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{6\times 5} \\ 0_{6\times 5} & \cdots & 0_{6\times 5} & \mathsf{P}_n \end{pmatrix}. \tag{9.88}$$

The unknown vectors are chosen such that $\hat{\mathbf{d}}_I^\alpha \cdot \mathbf{d}_I = 0$ holds for $\alpha \in \{1, 2\}$. Because of that and (9.73), the columns of the matrix P span the null space of the constraint matrix

$$\mathsf{G}(\mathbf{q})\mathsf{P}(\mathbf{q}) = \mathsf{P}^T(\mathbf{q})\mathsf{G}^T(\mathbf{q}) = 0. \tag{9.89}$$

We reduce the dimension of the system to $6n$ by multiplying (9.74) by P^T from the left:

$$\mathsf{P}^T(\mathbf{q})\,(\mathbf{R}(\mathbf{q}) - \mathbf{F}) = \mathbf{0}, \tag{9.90}$$

$$\Lambda(\mathbf{q}) = \mathbf{0}. \tag{9.91}$$

By eliminating the constraint forces also the Lagrangiang multipliers disappear. Hence, the degrees of freedom per node reduce to six. To get rid of the constraint equation (9.91), a local[7] parametrization $\mathbf{T}\colon \mathbb{R}^5 \to \mathbb{R}^6$ of the state vector $\mathbf{q} = \mathbf{T}(\mathbf{u})$ is introduced. This function should guarantee the unit length of the director in every node of the finite element mesh or more general

$$\Lambda(\mathbf{T}(\mathbf{u})) = 0 \quad \forall \mathbf{u} \in \mathbb{R}^{5n}. \tag{9.92}$$

The parametrization depends on the states $\mathbf{p}$ of the last time or increment step according to weather a dynamic or static problem has to be solved. Restricted to one single node, the function $\mathbf{T}$ is a translation for the midsurface point and a rotation for the director:

[7] It is called local, because the parametrization differs from time step to time step.

$$\mathbf{T}(\mathbf{u}) = \begin{array}{l} \varphi_{\mathbf{q}} = \varphi_{\mathbf{p}} + \mathbf{u}_{\varphi} \\ \mathbf{d}_{\mathbf{q}} = \exp(\widehat{u_{\mathbf{d}^1}\mathbf{d}^1_{\mathbf{p}} + u_{\mathbf{d}^2}\mathbf{d}^2_p})\mathbf{d}_{\mathbf{p}}. \end{array} \tag{9.93}$$

The function $\exp : so(3) \to S0(3)$ is the classical Rodriguez formula, which can be found in [17] for example. It generates a matrix, which represents a rotation around the vector $u_{\mathbf{d}^1}\mathbf{d}^1_{\mathbf{p}} + u_{\mathbf{d}^2}\mathbf{d}^2_p$ about the angle $\|u_{\mathbf{d}^1}\mathbf{d}^1_{\mathbf{p}} + u_{\mathbf{d}^2}\mathbf{d}^2_p\|$. The vectors $\mathbf{d}^{\alpha}_p$ are chosen such that together with the director $\mathbf{d}_{\mathbf{p}}$, they form an orthonormal system. With the parametrization $\mathbf{T}$, there are only $5n$ nonlinear equations left

$$\mathsf{P}^T(\mathbf{T}(\mathbf{u}))\left(\mathbf{R}\Big(\mathbf{T}(\mathbf{u})\Big) - \mathbf{F}\right) = \mathbf{0}. \tag{9.94}$$

These equations determine the parameters $\mathbf{u} \in \mathbb{R}^{5n}$, and so the solution of (9.90)–(9.91), (9.86)–(9.87) or (9.74)–(9.75). For more details and a verification of the method we refer to [2, 3, 13] and references therein.

Remark 3 Due to the parametrization of the director by a local rotation, the tangential matrix of the system has eigenvalues close to the imaginary axis. Therefore, the BDF-method is restricted to order two, if we want to assure the stability of the time integration scheme.

9.5 Coupling with MBS

To do a vehicle simulation, the discrete shell model representing the tire has to be coupled to the multibody system in a proper way. One possibility to achieve this could be a *monolithic* approach to time integration as introduced in [2]. Here, the multibody system and the flexible shell are integrated as one fully coupled system by a single time integration scheme.

As alternative one could choose a *co-simulation* approach, which often is the preferred variant within industrial applications [10]. Here, the vehicle and the tire are integrated as separated systems. At certain times these are synchronized in the following way: The rigid body defining the rim acts as an interface between the two systems. The vehicle determines its position, orientation and corresponding velocity at the synchronization time. With that information the tire is integrated up to this point and delivers the forces and torques acting on the rim.

With these inputs the multibody system integrates to the next synchronization time and in this way determines implicitly the new kinematic state of the rim. Then again the tire integrates afterwards and so on. This leads to three different time step sizes:

τ_{rigid} the step size of the rigid body system,
τ_{shell} the step size of the discrete shell,
τ_{sync} the step size of the synchronization times.

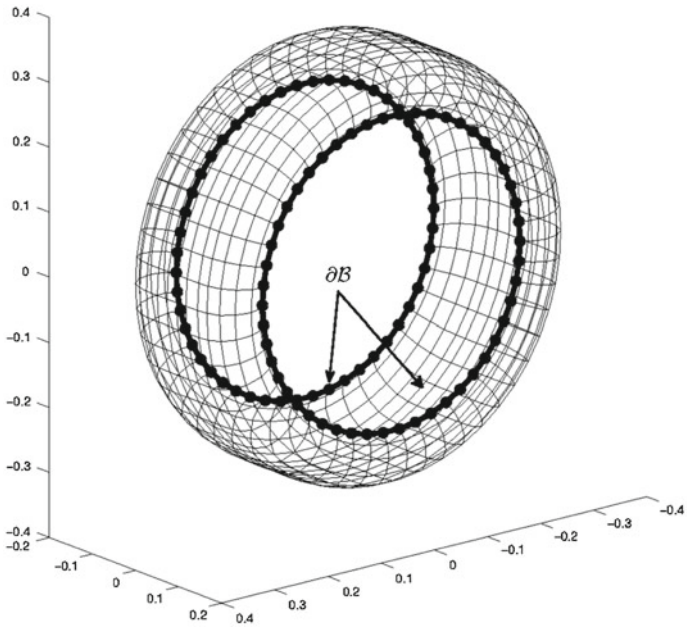


Fig. 9.5 Moving Dirichlet boundary

The time step of the discrete shell system is restricted such that an integer multiple $k \in \mathbb{N}$ is equal to the time step of the synchronization $\tau_{sync} = k\tau_{shell}$. For simplicity we assume here that $k = 1$, which means that only one time step is integrated between the synchronization.[8]

Because we are focused on the incorporation of the shell, we are not interested in the multibody system and its time step size τ_{rigid}. The only thing needed from the multibody is the kinematic state at the old t_n and new t_{n+1} synchronization times. Through these two quantities we get a displacement $\Delta\mathbf{r}$ and a rotation matrix $\Delta\mathsf{R}$ representing the rigid body motion.

To transfer the motion of the rim to the discrete shell representing the tire, we assume that its boundary $\partial\mathscr{B}$ (see Fig. 9.5) is fixed to the rim. Therefore the position and the director of the nodes are prescribed through the time interval $[t_n, t_{n+1}]$. Therefore the global state vector of the nodes is split into free and predefined part

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}^{free} \\ \mathbf{q}^{pre} \end{pmatrix} \quad \lambda = \begin{pmatrix} \lambda^{free} \\ \mathbf{0} \end{pmatrix}, \tag{9.95}$$

which contain the corresponding states of the nodes.

To do the time step we have to solve the nonlinear problem from (9.80)–(9.82). For simplicity we omit indexes and look for a solution of the corresponding static

[8] If $k \neq 1$ the kinematic states must be interpolated in the interval $[t_n, t_{n+1}]$.

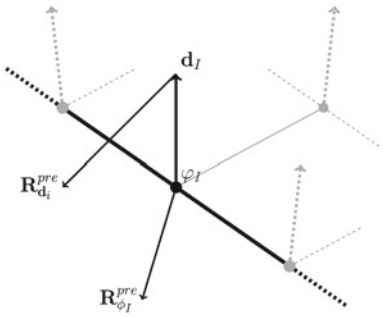


Fig. 9.6 Local forces and torques on a boundary node

problem (9.86)–(9.87) with additional forces due to the dynamics of the system. Because of the Dirichlet boundary, the variation of the prescribed nodes $\delta\mathbf{q}^{pre} = 0$ vanishes. Therefore the number of equations reduces to the number of free nodes

$$\mathbf{R}^{free}(\mathbf{q}, \lambda) = 0, \tag{9.96}$$

$$\Lambda^{free}(\mathbf{q}) = 0. \tag{9.97}$$

Note that the last equation is solved with respect to $\mathbf{q}^{free}$ and λ^{free}, but the system also depends on the values $\mathbf{q}^{pre}$. We also get the forces with respect to the vanishing variations $\mathbf{R}^{pre}(\mathbf{q})$ as by-product of the FE solution process of (9.96)–(9.97). This vector valued function is the assembly of the local forces $\mathbf{R}^{Pre}_I$ acting on the boundary nodes. Again this vector is decomposed in a position and a director force

$$\mathbf{R}^{Pre}_I = \begin{pmatrix} \mathbf{R}^{pre}_{\varphi_I} \\ \mathbf{R}^{pre}_{\mathbf{d}_I} \end{pmatrix} \in \mathbb{R}^6. \tag{9.98}$$

Figure 9.6 illustrates the two local forces. Calculating the global force acting on the center of the rim is done by summing up all local position forces

$$\mathbf{F}_{rim} = \sum_{\mathbf{q}_I \in \partial\mathscr{B}} \mathbf{R}^{pre}_{\varphi_I}(\mathbf{q}). \tag{9.99}$$

Because $\mathbf{d}_I$ acts as a lever arm for the local director force, a local torque $\mathbf{d}_I \times \mathbf{R}^{pre}_{\mathbf{d}_I}$ appears. This term together with the torque generated by the position forces and the distance to the center of the rim summing up to the global torque acting on the rim

$$\mathbf{T}_{rim} = \sum_{\mathbf{q}_I \in \partial\mathscr{B}} \left(\mathbf{d}_I \times \mathbf{R}^{pre}_{\mathbf{d}_I} + (\phi_I - \mathbf{r}) \times \mathbf{R}^{pre}_{\phi_I} \right). \tag{9.100}$$

9.6 Pressure and Contact

In addition to external loadings **B**, which are often asssumed to be configuration independent (like gravity), we have to incorporate specific configuration dependent loads. If we want to simulate a tire we have to deal with one sided contact with a road surface and pressure load. In the first section we introduce a penalty method to handle normal (frictionless) contact with a rigid road surface. In the second one we follow the work of [18] to incorporate loads due to inflation pressure.

9.6.1 Normal Contact

The tire has to interact with a rigid road, which is given via a height profile $\mathbf{p} = (x, y, h(x, y))^T$. Also its first two derivatives are analytically given. The rigidity of the road and its explicit description substantially simplifies the problem, compared to contact problems of two elastic discrete bodies.

The second contact partner is the outer surface of the shell continuum, which is given by $\bar{\mathbf{x}}(\chi) := \varphi(\chi) + \frac{h}{2}\mathbf{d}(\chi)$. For every point in this surface there exists a associated point in the road which has minimal distance $\hat{\mathbf{p}} = \text{argmin}\|\mathbf{p} - \bar{\mathbf{x}}\|$. Having these minimal distance point $\hat{\mathbf{p}}$, we define the penetration function by

$$g(\bar{\mathbf{x}}) = \begin{cases} 0, & \text{if}(\bar{\mathbf{x}} - \hat{\mathbf{p}}) \cdot \hat{\mathbf{n}} > 0, \\ (\bar{\mathbf{x}} - \hat{\mathbf{p}}) \cdot \hat{\mathbf{n}}, & \text{else.} \end{cases} \tag{9.101}$$

Here, the vector $\hat{\mathbf{n}}$ is the normal of the road surface in $\hat{\mathbf{p}}$. The variation of (9.101) in case of penetration ($g < 0$) is given by

$$\delta g = (\delta\phi + \frac{h}{2}\delta\mathbf{d}) \cdot \mathbf{n} = \begin{pmatrix} \mathbf{n} \\ \frac{h}{2}\mathbf{n} \end{pmatrix} \cdot \begin{pmatrix} \delta\varphi \\ \delta\mathbf{d} \end{pmatrix}. \tag{9.102}$$

With a positive scalar value $\varepsilon > 0$, a penalty term [20] is added to the balance of linear momentum (9.32)

$$G_c(\phi, \delta\phi) = \varepsilon \int_\omega g(\chi)\delta g(\chi) \mathrm{d}\chi. \tag{9.103}$$

Inserting the discretization (9.64) into (9.103) yields an additional force

$$\mathbf{R}_I^c(\mathbf{q}) \cdot \delta\mathbf{q}_I = \varepsilon \begin{pmatrix} \int_\omega g(\chi) N^I \mathrm{d}\chi \\ \frac{h}{2}\int_\omega g(\chi) N^I \mathrm{d}\chi \end{pmatrix} \cdot \begin{pmatrix} \delta\varphi_I \\ \delta\mathbf{d}_I \end{pmatrix} \tag{9.104}$$

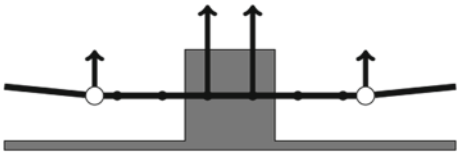


Fig. 9.7 Contact evaluation inside the element (*STS*). Small cleat is detected

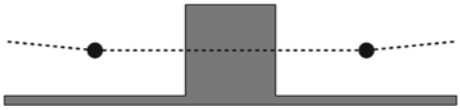


Fig. 9.8 Contact evaluation at the node of the FE-mesh (*NTS*). Small cleat is *not* detected

acting on each node of the discrete shell. These local contact forces are assembled to a global contact force $\mathbf{R}^c = \left(\mathbf{R}_1^c, \ldots, \mathbf{R}_n^c\right)^T$. The integrals of (9.104) are evaluated by numerical integration inside the finite elements. This leads to a surface to surface (STS) contact, with the road surface given analytically. Therefore we can increase the resolution of contact independent of the finesse of the FE-mesh.

Instead of evaluating (9.104) inside the elements, the contact force could be computed only in the nodes of the finite element mesh. In the literature, this approach is often called node to surface (NTS) contact. Here the finesse of the FE mesh determines the resolution of the contact. The local contact force from (9.104) becomes

$$\mathbf{R}_I^c(\mathbf{q}) = \varepsilon \begin{pmatrix} g(\bar{\mathbf{x}}_I) \\ \frac{h}{2} g(\bar{\mathbf{x}}_I) \end{pmatrix}, \tag{9.105}$$

with $\bar{\mathbf{x}}_I := \varphi_I + \frac{h}{2}\mathbf{d}_I$. Because in tire simulation we often deal with small cleats, the (NTS) method would lead to very fine meshes. This would decrease the computational speed in an unfavourable way.

Figure 9.7 illustrates the benefit of the STS method: The contact with a cleat which is smaller than the size of an element is detected. In contrast to that if the NTS method is used with the same element and cleat, no contact is detected (see Fig. 9.8).

9.6.2 Pressure Load

A tire is inflated with a certain pressure, which is assumed to be spatially constant within the cavity volume. For this reason, an additional force arises at the surface pointing into normal direction. The incorporation of a pressure load into a 3D continuum based FE concept is presented in [18]. This approach is adopted for the midsurface, which yields a contribution to the weak form

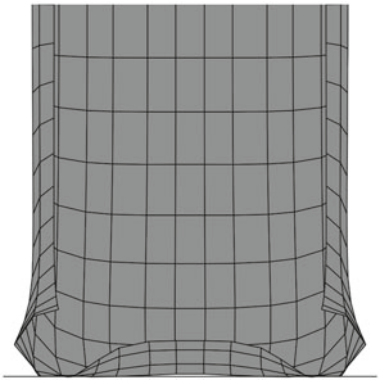


Fig. 9.9 Deflection of 35 mm relative to the first contact, no inflation pressure

$$G_p(\phi, \delta\phi) = \int_{\mathscr{S}} p\mathbf{a}_3 \cdot \delta\varphi \mathrm{d}\mathbf{x} = \int_{\omega} p(\mathbf{a}_1 \times \mathbf{a}_2) \cdot \delta\varphi \mathrm{d}\chi. \tag{9.106}$$

Note that the integral is taken over the deformed configuration $\mathscr{S}$, to create the correct force. By inserting the discretization from (9.64) in (9.106), the discrete version of the pressure term reads

$$G_p(\varphi^h, \delta\varphi^h) = \sum_I \sum_J \sum_K \delta\varphi_K \cdot (\varphi_I \times \varphi_J) \int_{\omega} N^I_{,1} N^J_{,2} N^K \, \mathrm{d}\mathbf{x}. \tag{9.107}$$

The resulting force depends on the current discrete configuration φ_I. As a consequence, for the solution procedure the contribution to the tangential matrix

$$K^p_{IJ} = \begin{pmatrix} \sum_K \hat{\varphi}_K \int_{\omega} \left(N^I_{,1} N^J_{,2} - N^I_{,2} N^J_{,1} \right) \mathrm{d}\chi \; 0_{3\times 3} \\ 0_{3\times 3} \qquad\qquad\qquad 0_{3\times 3} \end{pmatrix} \tag{9.108}$$

has to be computed, with a skew symmetric matrix $\hat{\varphi}_K \in \mathbb{R}^{3\times 3}$, such that for all $\mathbf{x} \in \mathbb{R}^3$ the equality $\hat{\varphi}_K \mathbf{x} = \varphi_K \times \mathbf{x}$ holds.

Together with the normal contact formulation from Sect. 9.6.1, a static simulation is performed as a first test. The discrete model is flattened out against a flat surface. In Fig. 9.9 no pressure is present inside the wheel, which induces bending in the tread. The contact region is a small banded ellipse around the ideal contact patch. This phenomenon is also known in practice in case of under-inflation. In Fig. 9.10, the same simulation is shown with 2 *bar* inflation pressure. Hence, the bending vanishes and the tread is completely in contact with the road around the ideal contact patch.

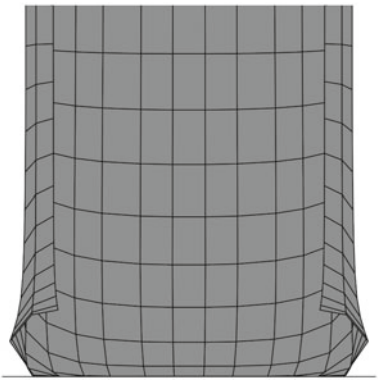


Fig. 9.10 Deflection of 35 mm relative to the first contact, 2 bar inflation pressure

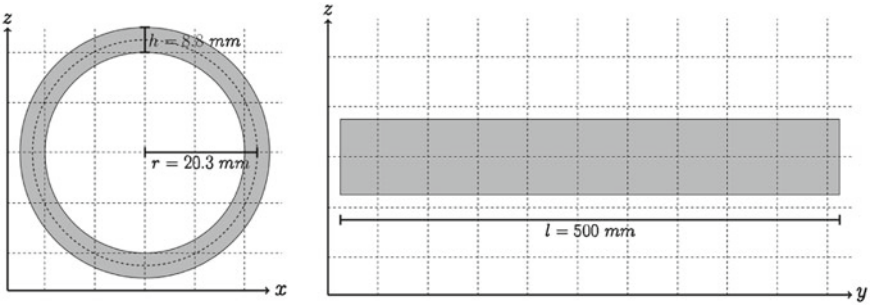


Fig. 9.11 Radius, thickness and length of the cylinder

9.7 Numerical Examples

In this section we show some numerical examples. Because we have not included frictional contact yet, only quasi static simulations are performed. In the first section we verify the orthotropic material law from Sect. 9.3.2, by testing it against a fully three dimensional simulation. In the second section the flattening of tire modeled by a discrete shell is simulated and compared to real measurements.

9.7.1 Bending of an Orthotropic Cylinder

To verify the orthotropic material law developed in Sect. 9.3, we compare our shell element against an *Abaqus* orthotropic eight node brick element. As a benchmark example the bending of a cylinder is chosen, with the geometry given in Fig. 9.11. The material is assumed as orthotropic and homogeneous along the cylinder, with

Table 9.3 Material data of the cylinder in 10^6 Pa, with $\mathbf{E}_1$ pointing into lateral, $\mathbf{E}_2$ into circumferential and $\mathbf{E}_3$ in thickness direction

E_1	E_2	E_3	ν_{12}	ν_{13}	ν_{23}	G_{12}	G_{13}	G_{23}
48.4	26.9	26.9	0.26	0.26	0.48	25	25	9.55

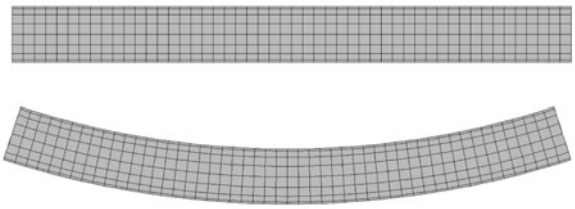


Fig. 9.12 At *top* reference configuration, *below* the bended deformed configuration

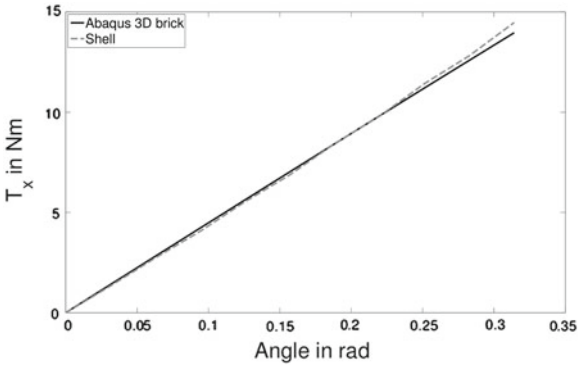


Fig. 9.13 Comparison of the 3D brick element against the discrete shell

its coefficients given in Table 9.3. The shell and the three dimensional model are discretized with the same number of elements in longitudinal and circumferential direction. The *Abaqus* model is additionally discretized in thickness direction. The simulation is performed quasi static, by fixing both ends of the cylinder to a rigid body and rotating these step-wise with the angle Θ respectively $-\Theta$ (see Fig. 9.12). Through the imposed motion of the boundary, a reaction torque arises, which is plotted against Θ in Fig. 9.13. In this plot it can be observed that the results of the 3D brick element and our shell element match quite well. This illustrates that the transformation of the orthotropic three dimensional material law to a shell material law as developed in Sect. 9.3.2 works properly.

Table 9.4 Material data for the simulation

	E_1 (MPa)	E_2 (MPa)	E_3 (MPa)	ν_{12}	ν_{13}	ν_{23}	G_{12} (MPa)	G_{13} (MPa)	G_{23} (MPa)
Tread	150	10	30	0.1	0.4	0.4	36.4	32.1	7.14
Sidewall	17	19	17	0.2	0.4	0.4	7.5	6.07	6.43

N_1 points into circumferential, N_2 in lateral direction

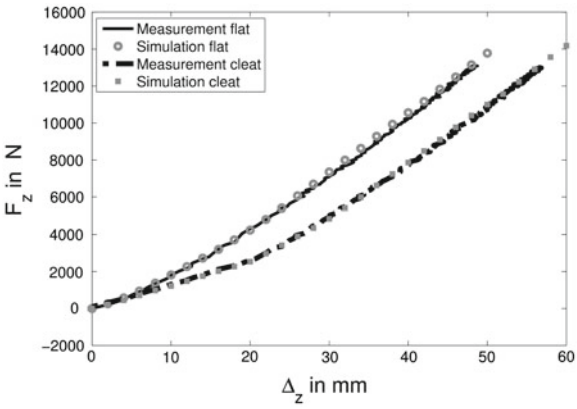


Fig. 9.14 Results of the simulation plotted against measurements

9.7.2 Flattening Out a Tire

Pressing the inflated tire against a flat rigid ground is the most simple experiment realized in a real physical test rig. Hereby, the stiffness of the tire is measured, which relates vertical deflection against the resulting force. The same test is done, with a cleat mounted on the flat surface, see Fig. 9.15. Because the tire is not rolling, only normal contact is needed in this case. These two tests are simulated with the discrete shell model described above.

As comparison a radial passenger car tire with 17 inch.[9] rim diameter and 225 mm width is chosen. The inflation pressure is 2.5 bar for both tests. To built up the FE mesh, a cross section is discretized with a number of points representing the midsurface. By revolving these nodes around the center line of the rim, a 2D quadrilateral mesh is constructed representing the undeformed mid surface of the tire. The condition of two parallel edges, made in Sect. 9.4.1 to enable a simple representation (by a single angle quantity) of orthotropic materials at the discrete level, is fulfilled. In every node of the mesh a unit vector is needed, representing the discrete director. These are generated heuristically, in a way that they are approximately normal to the midsurface.

[9] The units do not match because of the usual standardization.

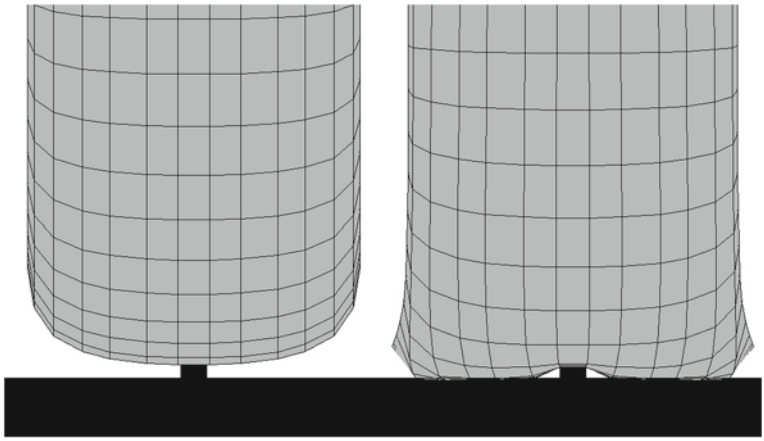


Fig. 9.15 Flattening out against a longitudinal cleat

In Fig. 9.14, a comparison between simulation results and measurements is shown. The deflection is plotted against the resulting force acting in the rim center. As mentioned above, two different simulations are performed. In both cases the orhotropic material parameters from Table 9.4 are used.

The inhomogeneities are approximated by taking different parameters per element in the tread and the sidewall. The material data from Table 9.4 are adjusted such that the simulation results match with the measurements in Fig. 9.14. Note that the discrete shell parametrized by this approach works for both simulations.

9.8 Conclusions and Outlook

In this work we have developed a tire model based on geometrically exact shells, which (according to our knowledge) is the first tire model of this kind. With this model we are able to handle isotropic as well as three dimensional orthotropic material data. The discrete shell is able to interact with a multibody system via co-simulation. Also the tire specific loads like normal contact or inflation pressure have been implemented in our model. We have shown that our tire model can be parametrized such that the results of a real physical experiment are reproduced by a simulation. However, it would be desirable to predict the material data from the three-dimensional geometry of the inner reinforcement structure of the tire and the elastic material data of the various materials. To compare our tire model against a dynamic simulation, we have to handle frictional contact with the rigid road surface. These two problems will be the topics of our further work.

Appendix: Material Coefficients for the Shell

The *isotropic* coefficients for the shell material law are given by:

$$\mathbb{H}_{\text{iso}}^{\alpha\alpha\alpha\alpha} = \frac{E}{1-\nu^2} G^{\alpha\alpha} G^{\alpha\alpha}, \tag{9.109}$$

$$\mathbb{H}_{\text{iso}}^{\alpha\alpha\beta\beta} = \frac{E}{1-\nu^2} \left((1-\nu) G^{\alpha\beta} G^{\alpha\beta} + \nu G^{\alpha\alpha} G^{\beta\beta} \right), \tag{9.110}$$

$$\mathbb{H}_{\text{iso}}^{\alpha\beta\alpha\beta} = \frac{E}{1-\nu^2} \left((1+\nu) G^{\alpha\beta} G^{\alpha\beta} + (1-\nu) G^{\alpha\alpha} G^{\beta\beta} \right), \tag{9.111}$$

$$\mathbb{H}_{\text{iso}}^{\alpha\alpha\alpha\beta} = \frac{E}{1-\nu^2} G^{\alpha\alpha} G^{\alpha\beta}, \tag{9.112}$$

$$\hat{\mathbb{H}}_{\text{iso}}^{\alpha\beta} = \frac{E}{2(\nu+1)} G^{\alpha\beta}. \tag{9.113}$$

The coefficients of the *orthotropic* shell material law are given by:

$$\mathbb{H}_{\text{orth}}^{\alpha\alpha\alpha\alpha} = \sum_{\iota=1}^{2} \left(L_1^{\iota\iota} - \frac{L_1^{3\iota}}{L_1^{33}} \right) (B_\iota^\alpha)^4 + 2(L_1^{12} + 2L_2^{12})(B_1^\alpha B_2^\alpha)^2, \tag{9.114}$$

$$\mathbb{H}_{\text{orth}}^{\alpha\beta\alpha\beta} = \sum_{\iota=1}^{2} \left(L_1^{\iota\iota} - \frac{L_1^{3\iota}}{L_1^{33}} \right) (B_\iota^\alpha B_\iota^\beta)^2 + 4L_1^{12} B_1^\alpha B_2^\alpha B_1^\beta B_2^\beta \tag{9.115}$$

$$+ L_2^{12}((B_1^\alpha B_2^\beta)^2 + (B_2^\alpha B_1^\beta)^2), \tag{9.116}$$

$$\mathbb{H}_{\text{orth}}^{\alpha\beta\alpha\beta} = \sum_{\iota=1}^{2} \left(L_1^{\iota\iota} - \frac{L_1^{3\iota}}{L_1^{33}} \right) (B_\iota^\alpha B_\iota^\beta)^2 + 2(L_1^{12} + L_2^{12})(B_1^\alpha B_2^\alpha B_1^\beta B_2^\beta) \tag{9.117}$$

$$+ L_2^{12}((B_1^\alpha B_2^\beta)^2 + (B_2^\alpha B_1^\beta)^2), \tag{9.118}$$

$$\mathbb{H}_{\text{orth}}^{\alpha\alpha\alpha\alpha} = \sum_{\iota=1}^{2} \left(L_1^{\iota\iota} - \frac{L_1^{3\iota}}{L_1^{33}} \right) (B_\iota^\alpha)^3 B_\iota^\beta \tag{9.119}$$

$$+ (L_1^{12} + 2L_2^{12})((B_1^\alpha)^2 B_2^\alpha B_2^\beta + (B_2^\alpha)^2 B_1^\alpha B_1^\beta, \tag{9.120}$$

$$\hat{\mathbb{H}}_{\text{orth}}^{\alpha\beta} = \sum_{\iota=1}^{2} L_2^{\iota 3} 2 B_\iota^\alpha B_\iota^\beta. \tag{9.121}$$

Note that we make use of the approximation $G^{\alpha\beta} \approx A^{\alpha\beta}$, such that the integration in the transverse (thickness) direction can be done analytically.

References

1. Bauchau OA, Choi J-Y, Bottasso CL (2002) On the modeling of shells in multibody dynamics. Multibody Sys Dyn 8:459–489
2. Betsch P, Sänger N (2009) On the use of geometrically exact shells in a conserving framework for flexible multibody dynamics. Comput Methods Appl Mech Eng 198:1609–1630
3. Betsch P, Leyendecker S (2006) The discrete null space method for the energy consistent integration of constrained mechanical systems. Part II: multibody dynamics. Int J Numer Methods Eng 67:499–552
4. Bonet J, Wood RD (2008) Nonlinear continuum mechanics for finite element analysis. Cambridge University Press, Cambridge
5. Brank B, Perić D, Damjanić FB (1995) On implementation of a nonlinear four node shell finite element for thin multilayered elastic shells. Comput Mech 7:513–525
6. Chapelle D, Bathe K-J (2011) The finite element analysis of shells—Fundamentals, 2nd edn. Springer, Berlin
7. do Carmo MP (1976) Differential geometry of curves and surfaces. Prentice Hall, Englewood Cliffs
8. Dvorkin EN, Bathe KJ (1984) A continuum mechanics based four-node shell element for general nonlinear analysis. Comput Struct 1:77–88
9. Ericksen JL, Truesdell C (1958) Exact theory of stress and strain in rods and shells. Arch Ration Mech Anal 1:295–323
10. Gallrein A, Bäcker M, Gizatullin A (2013) Structural MBD tire models: closing the gap to structural analysis—History and fture of parameter identification SAE technical paper, 2013–01-0630
11. Itskov M (2001) A generalized orthotropic hyperelastic material model with application to incompressible shells. Int J Numer Methods Eng 50:1777–1799
12. Jones R (1998) Mechanics of composite materials. CRC Press, Boca Raton
13. Leyendecker S, Betsch P, Steinmann P (2008) The discrete null space method for the energy consistent integration of constrained mechanical systems. Part III: flexible multibody dynamics. Multibody Sys Dyn 19:45–72
14. Lugner P, Plöchel M (2005) Tyre model performance test: first experiences and results. Vehicle Syst Dyn 45(1):48–62
15. Pacejka HB, Bakker E (1992) The magic formular tyre model. Vehicle Syst Dyn 21(1):1–18
16. Poldneff MJ, Heinstein MW (2010) Computational mechanics of rubber and tires. In: Gujrato PD, Leonov AI (eds) Modeling and simulation in polymers. Wiley-VCH, pp 385–403
17. Simo JC, Fox DD (1989) On a stress resultant geometrically exact shell model. Part I: formulation and optimal parametrization. Comput Methods Appl Mech Eng 72:267–304
18. Simo JC, Taylor RL, Wriggers P (1991) A note on finite-element implementation of pressure boundary loading. Commun Appl Numer Methods 16:341–359
19. Wanner G, Hairer E (1991) Solving ordinary differential equations II. Springer, Berlin
20. Wriggers P (2006) Computational contact mechanics. Springer, Berlin
21. Wriggers P (2008) Nonlinear finite element methods. Springer, Berlin

Chapter 10
Application of a Gyrostatic Rigid Body Formulation in the Context of a Direct Transcription Method for Optimal Control in Multibody Dynamics

Christian Becker and Peter Betsch

Abstract This chapter describes a new rigid body formulation for multibody dynamics, embedded in a natural coordinate framework, and its application as a control input interface for optimal control theory within a direct transcription method. In contrast to the rotationless formulation, see also [3], this scheme exploits the rotational invariance of certain rigid bodies, i.e. rotors, in regard to their mass distribution. This makes possible a separation of the general rigid body movement in a spin part and a complementary rotation part. The resulting equations of motion feature a simple mass matrix with mostly constant entries and without the need of transcendental functions. Furthermore the separated spin coordinate serves as a control input interface within an optimal control problem.

10.1 Introduction

The present work deals with the optimal control of multibody systems in terms of natural coordinates (NCs), ultimately leading to equations of motion in form of differential-algebraic equations (DAEs). This approach enables (i) the systematic implementation of open-loop and closed-loop multibody systems, (ii) the design of structure-preserving integrators, and (iii) the direct link to nonlinear finite element methods. However, the incorporation of necessary control inputs in the context of the optimal control problem presents a major challenge due to the lack of conjugated joint-coordinates. So far, feasible solutions to this task were either the direct incorporation of joint forces conjugated to NCs, see also [1, 8], or by adjoining

C. Becker · P. Betsch (✉)
Institute of Mechanics, Karlsruhe Institute of Technology (KIT),
Otto-Ammann-Platz 9, 76131 Karlsruhe, Germany
e-mail: christian.becker@kit.edu

P. Betsch
e-mail: peter.betsch@kit.edu

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_10,

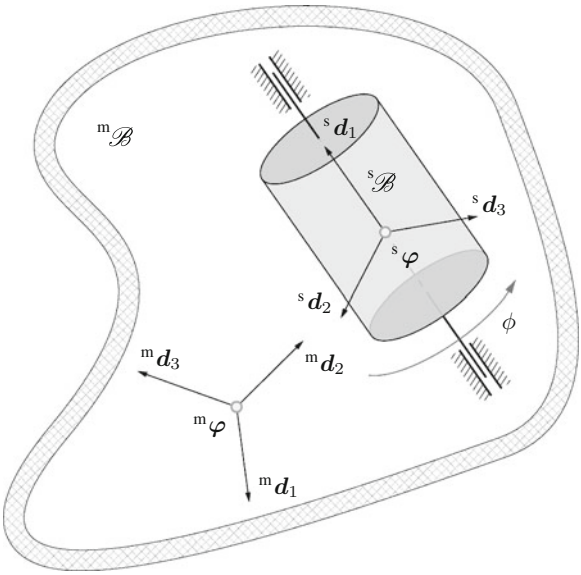


Fig. 10.1 Two body compound

augmented coordinates via additional constraints, see [4]. A further possibility is the present approach, namely a gyrostatic formulation, especially suited for rigid bodies with high rotational spin, such as rotors. Entrenched in the rotational parametrization of rigid bodies in terms of NCs, this scheme can be briefly described as the NC complement of the modified dynamical Euler's equations, to be found in classical mechanics, see also [6, 10].

The transition to the actual optimal control problem will be accomplished by the direct transcription of the equations of motion, see [5, 8, 9, 11]. The application of the proposed scheme will be demonstrated with selected numerical examples, where the goal is to minimize the overall control effort for a rest-to-rest maneuver.

10.2 Kinematics

For the sake of simplicity we will deal with a two body compound within the proposed formulation, see Fig. 10.1. Here, we denote the body, providing the translational and rotational coordinates in terms of NCs, as the *master* and corresponding quantities as $^{\mathsf{m}}\bullet$. The rigid body with symmetrical mass distribution and a relative axis of revolution along the axis of symmetry, i.e. the rotor, will be identified as the *slave* with the associated quantities $^{\mathsf{s}}\bullet$.

The spatial position vector for an arbitrary material point of the respective rigid body α is given by

$$ {}^{\alpha}\boldsymbol{x} = {}^{\alpha}\boldsymbol{\varphi} + {}^{\alpha}\boldsymbol{R}\,{}^{\alpha}\mathcal{X} \tag{10.1} $$

with $\alpha \in \{\mathsf{m}, \mathsf{s}\}$, where ${}^{\alpha}\mathcal{X}$ denotes the convective position, ${}^{\alpha}\boldsymbol{R} \in \{\mathcal{SO}(3) \,|\, {}^{\alpha}\boldsymbol{R} = {}^{\alpha}\boldsymbol{d}_i \otimes \boldsymbol{e}_i,\ {}^{\alpha}\boldsymbol{d}_i \cdot {}^{\alpha}\boldsymbol{d}_j = \delta_{ij}\}$ the rotational and ${}^{\alpha}\varphi$ the translational displacement of the respective material point.

In case of the master, the nine entries of its rotational tensor can be identified as direction cosines, which play the role of a redundant set of coordinates to describe the orientation of the associated body-fixed director triad $\{{}^{\mathsf{m}}\boldsymbol{d}_i\}$ in space. Consequently this set of coordinates is subject to independent constraints $\boldsymbol{\Phi} \in \mathbb{R}^m,\ m = 6$ enforcing the orthonormality of the triad, given by

$$ \Phi_i = {}^{\mathsf{m}}\boldsymbol{d}_j \cdot {}^{\mathsf{m}}\boldsymbol{d}_k - \delta_{jk} \tag{10.2} $$

In case of the slave body, the set $\{{}^{\mathsf{m}}\varphi, {}^{\mathsf{m}}R\}$ fulfills the purpose of a subspace of coordinates to describe the slave's translational and rotational displacement. The complete rotational description of the slave is defined by considering a subsequent follower rotation $\exp\hat{\boldsymbol{\phi}} \in \mathcal{SO}(3)$, where the angular coordinate $\phi = ||\mathrm{vect}(\hat{\boldsymbol{e}}_1\phi)||$ can be associated with the admissible relative twist between the slave and the master body. Thus, in respect to the slave, (10.1) yields

$$ \begin{aligned} {}^{\mathsf{s}}\boldsymbol{x} &= {}^{\mathsf{s}}\boldsymbol{\varphi} + {}^{\mathsf{s}}\boldsymbol{R}\,{}^{\mathsf{s}}\mathcal{X} \\ &= {}^{\mathsf{m}}\boldsymbol{\varphi} + {}^{\mathsf{m}}\boldsymbol{R}\boldsymbol{c} + {}^{\mathsf{m}}\boldsymbol{R}\boldsymbol{C}\exp\phi\hat{\boldsymbol{e}}_1\,{}^{\mathsf{s}}\mathcal{X} \end{aligned} \tag{10.3} $$

here $\boldsymbol{c} \in \mathbb{R}^3$ and $\boldsymbol{C} \in \mathcal{SO}(3)$ denote convective quantities in respect to the master triad. By collecting all geometrical degrees of freedom in the global configuration vector $\boldsymbol{q} \in \mathbb{R}^n,\ n = 13$, we can define the feasible configuration manifold Q and the conjugated tangent space $T_{\boldsymbol{q}}\mathsf{Q}$ of the constrained system as follows

$$ \mathsf{Q} = \left\{\boldsymbol{q} \in \mathbb{R}^n \,|\, \Phi_i(\boldsymbol{q}) = 0,\ 1 \le i \le m\right\} \qquad T_{\boldsymbol{q}}\mathsf{Q} = \left\{\boldsymbol{v} \in \mathbb{R}^n \,|\, \textstyle\sum_{i=1}^{m} \nabla\Phi_i(\boldsymbol{q}) \cdot \boldsymbol{v} = 0\right\} \tag{10.4} $$

10.3 Dynamics

10.3.1 Precondition on the Inertial Parametrization

The spatial distribution of mass of a rigid body $\mathscr{B}$ w.r.t. an arbitrary body-fixed director triad $\{d_i\}$ is completely parameterized by its mass $M_\varphi = \int_{\mathscr{B}} \rho_0 \,\mathrm{d}V \in \mathbb{R}$, static moment $\boldsymbol{s} = \int_{\mathscr{B}} \mathcal{X}\rho_0 \,\mathrm{d}V \in \mathbb{R}^3$ and Euler tensor $\boldsymbol{E} = \int_{\mathscr{B}} \mathcal{X} \otimes \mathcal{X}\rho_0 \,\mathrm{d}V \in \mathbb{R}^{3\times3}$. The parameterization of the master body is arbitrary, whereas preconditions need to

be imposed on the inertial parameters and the associated director triad of the slave body to avoid transcendental configuration dependency of the resulting mass tensor.

By choice of design we assume the director triad $\{^{\mathrm{s}}d_i\}$ to coincide with the principle axis frame, i.e. the conjugated Euler tensor features solely entries on the main diagonal and the conjugated static moment becomes zero. Furthermore, the mass distribution of the slave body needs to be invariant under arbitrary rotations $\exp \phi \hat{e}_1$. This holds for a set of rigid bodies, whose mass distribution features a $\infty-$fold symmetry axis along e_1. Symmetrical groups of this type can be defined by their Euler tensor in spectral decomposed form $\boldsymbol{E} = \lambda_i \boldsymbol{e}_i \otimes \boldsymbol{e}_i$, where (i) $\lambda_1 \neq \lambda_2 = \lambda_3$ characterize bodies with cylindrical and (ii) $\lambda_1 = \lambda_2 = \lambda_3$ bodies with spherical symmetry. As the more general case we choose group (i), hence the inertial parameters of the slave can be postulated as

$$^{\mathrm{s}}\boldsymbol{E} = E^1 \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \mathcal{E}\left(\boldsymbol{I} - \boldsymbol{e}_1 \otimes \boldsymbol{e}_1\right) \tag{10.5}$$

together with the rotational invariance along the axis of symmetry

$$^{\mathrm{s}}\boldsymbol{E} = \exp \phi \hat{\boldsymbol{e}}_1 \, ^{\mathrm{s}}\boldsymbol{E} \exp \phi \hat{\boldsymbol{e}}_1{}^T \tag{10.6}$$

10.3.2 Kinetic Energy

Based on the general formula $T = \frac{1}{2} \int_{\mathscr{B}} \rho_0 \dot{\boldsymbol{x}} \cdot \dot{\boldsymbol{x}} \, \mathrm{d}V$, the kinetic energy of an arbitrary rigid body yields

$$\begin{aligned} {}^{\alpha}T &= \frac{1}{2} \int_{\mathscr{B}} \rho_0 \left(\dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{R}}\mathcal{X}\right) \cdot \left(\dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{R}}\mathcal{X}\right) \mathrm{d}V \\ &= \frac{1}{2} \left[\dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}} \int_{\mathscr{B}} \rho_0 \, \mathrm{d}V + 2 \operatorname{tr} \left(\dot{\boldsymbol{R}} \int_{\mathscr{B}} \mathcal{X} \rho_0 \, \mathrm{d}V \otimes \dot{\boldsymbol{\varphi}} \right) + \operatorname{tr} \left(\dot{\boldsymbol{R}} \left(\int_{\mathscr{B}} \mathcal{X} \otimes \mathcal{X} \rho_0 \, \mathrm{d}V \right) \dot{\boldsymbol{R}}^T \right) \right] \\ &= \frac{1}{2} \left[M_{\varphi} \dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}} + 2 \operatorname{tr} \left(\dot{\boldsymbol{R}} \boldsymbol{s} \otimes \dot{\boldsymbol{\varphi}} \right) + \operatorname{tr} \left(\dot{\boldsymbol{R}}^T \dot{\boldsymbol{R}} \boldsymbol{E} \right) \right] \end{aligned} \tag{10.7}$$

By inserting the time derivative of (10.3) into (10.7) we obtain the kinetic energy of the rotor $^{\mathrm{s}}T$. Here, we treat translational, coupling and the spin contribution of the kinetic energy separately as follows. The translational contribution yields

$$\begin{aligned} {}^{\mathrm{s}}M_{\varphi} \, {}^{\mathrm{s}}\dot{\boldsymbol{\varphi}} \cdot {}^{\mathrm{s}}\dot{\boldsymbol{\varphi}} &= {}^{\mathrm{s}}M_{\varphi} \left(\dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{R}}\boldsymbol{c}\right) \cdot \left(\dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{R}}\boldsymbol{c}\right) \\ &= {}^{\mathrm{s}}M_{\varphi} \left[\dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}} + 2\dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{R}}\boldsymbol{c} + \operatorname{tr} \left(\dot{\boldsymbol{R}}\boldsymbol{c} \otimes \boldsymbol{c}\dot{\boldsymbol{R}}^T \right) \right] \end{aligned} \tag{10.8}$$

Due to the presumption $^{\mathrm{s}}\boldsymbol{s} = \int_{^{\mathrm{s}}\mathscr{B}} {}^{\mathrm{s}}\mathcal{X} \, \mathrm{d}V = \boldsymbol{0}$ the coupling contribution vanishes. Eventually the spin contribution is given by

$$\begin{aligned}\operatorname{tr}\left({}^{\mathrm{s}}\dot{\boldsymbol{R}}\,{}^{\mathrm{s}}\boldsymbol{E}\,{}^{\mathrm{s}}\dot{\boldsymbol{R}}^T\right) &= \operatorname{tr}\left(\left[\dot{\boldsymbol{R}}\boldsymbol{C}+\dot{\phi}\boldsymbol{R}\boldsymbol{C}\hat{\boldsymbol{e}}_1\right]\exp\phi\hat{\boldsymbol{e}}_1\,{}^{\mathrm{s}}\boldsymbol{E}\exp\phi\hat{\boldsymbol{e}}_1{}^T\left[\boldsymbol{C}^T\dot{\boldsymbol{R}}^T+\hat{\boldsymbol{e}}_1^T\boldsymbol{C}^T\boldsymbol{R}^T\dot{\phi}\right]\right)\\ &= \operatorname{tr}\Big(\dot{\boldsymbol{R}}\boldsymbol{C}\,{}^{\mathrm{s}}\boldsymbol{E}\boldsymbol{C}^T\dot{\boldsymbol{R}}^T+\dot{\phi}\boldsymbol{R}\boldsymbol{C}\hat{\boldsymbol{e}}_1\,{}^{\mathrm{s}}\boldsymbol{E}\boldsymbol{C}^T\dot{\boldsymbol{R}}^T+\dot{\boldsymbol{R}}\boldsymbol{C}\,{}^{\mathrm{s}}\boldsymbol{E}\hat{\boldsymbol{e}}_1^T\boldsymbol{C}^T\boldsymbol{R}^T\dot{\phi}\\ &\quad+\dot{\phi}^2\boldsymbol{R}\boldsymbol{C}\hat{\boldsymbol{e}}_1\,{}^{\mathrm{s}}\boldsymbol{E}\hat{\boldsymbol{e}}_1^T\boldsymbol{C}^T\boldsymbol{R}^T\Big)\end{aligned}\tag{10.9}$$

Here use has been made of the rotational invariance in (10.6). Furthermore we introduce ${}^{\mathrm{s}}\bar{\boldsymbol{E}}=\boldsymbol{C}\,{}^{\mathrm{s}}\boldsymbol{E}\boldsymbol{C}^T$ for the Euler tensor and $\hat{\boldsymbol{C}}_1=\boldsymbol{C}\hat{\boldsymbol{e}}_1\boldsymbol{C}^T$, $\boldsymbol{C}_1=\boldsymbol{C}\boldsymbol{e}_1$ w.r.t. the relative axis of rotation as a transformation of convective quantities from the slave frame to convective quantities in the master frame. Together with

$$\begin{aligned}\hat{\boldsymbol{e}}_1\,{}^{\mathrm{s}}\boldsymbol{E}&=\hat{\boldsymbol{e}}_1\left[E^1\boldsymbol{e}_1\otimes\boldsymbol{e}_1+\mathcal{E}\left(\boldsymbol{I}-\boldsymbol{e}_1\otimes\boldsymbol{e}_1\right)\right]\\ &=\mathcal{E}\hat{\boldsymbol{e}}_1\end{aligned}\tag{10.10}$$

we eventually obtain the following form of the spin contribution from (10.9)

$$\operatorname{tr}\left({}^{\mathrm{s}}\dot{\boldsymbol{R}}\,{}^{\mathrm{s}}\boldsymbol{E}\,{}^{\mathrm{s}}\dot{\boldsymbol{R}}^T\right)=\operatorname{tr}\left(\dot{\boldsymbol{R}}^T\dot{\boldsymbol{R}}\,{}^{\mathrm{s}}\bar{\boldsymbol{E}}\right)+2\dot{\phi}\mathcal{E}\operatorname{tr}\left(\dot{\boldsymbol{R}}^T\dot{\boldsymbol{R}}\hat{\boldsymbol{C}}_1\right)+2\mathcal{E}\dot{\phi}^2\tag{10.11}$$

Inserting the separate contributions from (10.8) and (10.10) into (10.7) together with the kinetic energy of the master body leads to the kinetic energy of the whole two-body compound, given by

$$\begin{aligned}T&={}^{\mathrm{s}}T+{}^{\mathrm{m}}T\\ &=\frac{1}{2}\left({}^{\mathrm{m}}M_\varphi+{}^{\mathrm{s}}M_\varphi\right)\dot{\boldsymbol{\varphi}}\cdot\dot{\boldsymbol{\varphi}}+\dot{\boldsymbol{\varphi}}\cdot\dot{\boldsymbol{R}}\left({}^{\mathrm{m}}\boldsymbol{s}+{}^{\mathrm{s}}M_\varphi\boldsymbol{c}\right)+\frac{1}{2}\operatorname{tr}\left(\dot{\boldsymbol{R}}^T\dot{\boldsymbol{R}}\left[{}^{\mathrm{m}}\boldsymbol{E}+{}^{\mathrm{s}}\bar{\boldsymbol{E}}+{}^{\mathrm{s}}M_\varphi\boldsymbol{c}\otimes\boldsymbol{c}\right]\right)\\ &\quad+\dot{\phi}\mathcal{E}\operatorname{tr}\left(\dot{\boldsymbol{R}}^T\boldsymbol{R}\hat{\boldsymbol{C}}_1\right)+\mathcal{E}\dot{\phi}^2\end{aligned}\tag{10.12}$$

Note that the fourth term or coupling term can alternatively be written as

$$\begin{aligned}\dot{\phi}\mathcal{E}\operatorname{tr}\left(\dot{\boldsymbol{R}}^T\boldsymbol{R}\hat{\boldsymbol{C}}_1\right)&=\dot{\phi}\mathcal{E}\operatorname{tr}\left(\hat{\boldsymbol{\Omega}}^T\hat{\boldsymbol{C}}_1\right)\\ &=2\mathcal{E}\dot{\phi}\boldsymbol{\Omega}\cdot\boldsymbol{C}_1\end{aligned}\tag{10.13}$$

with the convective angular velocity of the master body $\boldsymbol{\Omega}\in\mathbb{R}^3$.

10.3.3 Equations of Motion

Within Hamilton's principle, the action of a constrained mechanical system is defined by the functional

$$S = \int_{t_0}^{t_N} [L + \boldsymbol{\Phi} \cdot \boldsymbol{\lambda}] \, \mathrm{d}t \tag{10.14}$$

where we focus on the kinetic energy, i.e. $L = T$, and λ, $\boldsymbol{\Phi} \in \mathbb{R}^m$ denote the Lagrangian multipliers and conjugated constraints given by (10.2). Next, we apply a partitioned formulation of the action as follows

$$S(\boldsymbol{\varphi}(t), \boldsymbol{R}(t), \phi(t)) = S_{\mathsf{dir}}(\boldsymbol{\varphi}, \boldsymbol{R}) + S_{\mathsf{add}}(\boldsymbol{R}, \phi) + S_{\mathsf{con}}(\boldsymbol{R}) \tag{10.15}$$

where S_{dir} has the structure of the director formulation, see also [3],

$$S_{\mathsf{dir}}(\boldsymbol{\varphi}, \boldsymbol{R}) = \int_{t_0}^{t_N} \left[\frac{1}{2} M_{\varphi}^{*} \dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{R}} \boldsymbol{s}^{*} + \frac{1}{2} \operatorname{tr} \left(\dot{\boldsymbol{R}}^{T} \dot{\boldsymbol{R}} \boldsymbol{E}^{*} \right) \right] \mathrm{d}t \tag{10.16}$$

S_{add} is the additional contribution to the action functional

$$S_{\mathsf{add}}(\boldsymbol{R}, \phi) = \mathcal{E} \int_{t_0}^{t_N} \left[\dot{\phi} \operatorname{tr} \left(\dot{\boldsymbol{R}}^{T} \boldsymbol{R} \hat{\boldsymbol{C}}_1 \right) + \dot{\phi}^2 \right] \mathrm{d}t \tag{10.17}$$

and $S_{\mathsf{con}}(\boldsymbol{R})$ is the constraint contribution

$$S_{\mathsf{con}}(\boldsymbol{R}) = \int_{t_0}^{t_N} \boldsymbol{\Phi} \cdot \boldsymbol{\lambda} \, \mathrm{d}t \tag{10.18}$$

Note that the inertial quantities $\bullet^{*}$ coincide with those in (10.12) for the two body compound. By applying the calculus of variation and subsequent integration by parts, (10.16) yields after a tedious but straightforward calculation

$$\delta S_{\mathsf{dir}} = - \int_{t_0}^{t_N} \delta \boldsymbol{\varphi} \cdot \left(M_{\varphi}^{*} \ddot{\boldsymbol{\varphi}} + \ddot{\boldsymbol{R}} \boldsymbol{s}^{*} \right) \mathrm{d}t - \int_{t_0}^{t_N} \operatorname{tr} \left(\delta \boldsymbol{R} \left[\boldsymbol{s}^{*} \otimes \ddot{\boldsymbol{\varphi}} + \boldsymbol{E}^{*} \ddot{\boldsymbol{R}}^{T} \right] \right) + \delta S_{\mathsf{dir}}^{\mathsf{b.c.}} \tag{10.19}$$

where the boundary condition $\delta S_{\mathsf{dir}}^{\mathsf{b.c.}} = \left(\delta \boldsymbol{\varphi} \cdot \left(M_{\varphi}^{*} \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{R}} \boldsymbol{s}^{*} \right) + \operatorname{tr} \left(\delta \boldsymbol{R} \left[\boldsymbol{s}^{*} \otimes \dot{\boldsymbol{\varphi}} + \boldsymbol{E}^{*} \dot{\boldsymbol{R}}^{T} \right] \right) \right) |_{t_0}^{t_N}$ vanishes, since the variations at the fixed endpoints vanish, i.e. $\delta \boldsymbol{q}|_{t_0} = \delta \boldsymbol{q}|_{t_N} = 0$. Similarly the variation of the additional contribution yields

$$\delta S_{\text{add}} = -\mathcal{E} \int_{t_0}^{t_N} \delta\phi \left[2\ddot{\phi} + \operatorname{tr}\left(\ddot{\boldsymbol{R}}^T \boldsymbol{R} \hat{\boldsymbol{C}}_1 \right) \right] \mathrm{d}t + \mathcal{E} \int_{t_0}^{t_N} \operatorname{tr}\left(\delta \boldsymbol{R} \hat{\boldsymbol{C}}_1 \left(2\dot{\phi} \dot{\boldsymbol{R}}^T + \ddot{\phi} \boldsymbol{R}^T \right) \right) \mathrm{d}t + \delta S_{\text{add}}^{\text{b.c.}} \tag{10.20}$$

here once again the boundary condition $\delta S_{\text{add}}^{\text{b.c.}} = \mathcal{E} \left[\delta\phi \left(2\dot{\phi} + \operatorname{tr}\left(\dot{\boldsymbol{R}}^T \boldsymbol{R} \hat{\boldsymbol{C}}_1 \right) \right) - \dot{\phi} \operatorname{tr}\left(\delta \boldsymbol{R} \hat{\boldsymbol{C}}_1 \boldsymbol{R}^T \right) \right] |_{t_0}^{t_N}$ vanishes. Eventually the variation of the constraint contribution yields

$$\delta S_{\text{con}}(\boldsymbol{R}) = \int_{t_0}^{t_N} \frac{\partial \left(\boldsymbol{\Phi}(\boldsymbol{R}) \cdot \boldsymbol{\lambda} \right)}{\partial \boldsymbol{R}} : \delta \boldsymbol{R} \, \mathrm{d}t + \int_{t_0}^{t_N} \boldsymbol{\Phi}(\boldsymbol{R}) \cdot \delta \boldsymbol{\lambda} \, \mathrm{d}t \tag{10.21}$$

By collecting all terms from (10.19) (10.20) and (10.21) the continuous equations of motion read

$$\begin{aligned} M_{\varphi}^* \ddot{\boldsymbol{\varphi}} + \ddot{\boldsymbol{R}} \boldsymbol{s}^* &= \boldsymbol{0} \quad \forall \delta\boldsymbol{\varphi} \\ \ddot{\boldsymbol{R}} \boldsymbol{E}^* + \ddot{\boldsymbol{\varphi}} \otimes \boldsymbol{s}^* + \mathcal{E} \boldsymbol{R} \hat{\boldsymbol{C}}_1 \ddot{\phi} + 2\mathcal{E} \dot{\boldsymbol{R}} \hat{\boldsymbol{C}}_1 \dot{\phi} + \frac{\partial \left(\boldsymbol{\Phi}(\boldsymbol{R}) \cdot \boldsymbol{\lambda} \right)}{\partial \boldsymbol{R}} &= \boldsymbol{0} \quad \forall \delta\boldsymbol{R} \\ \boldsymbol{R}^T \ddot{\boldsymbol{R}} : \mathcal{E} \hat{\boldsymbol{C}}_1 + 2\mathcal{E} \ddot{\phi} &= 0 \quad \forall \delta\phi \\ \boldsymbol{\Phi}(\boldsymbol{R}) &= \boldsymbol{0} \quad \forall \delta\boldsymbol{\lambda} \end{aligned} \tag{10.22}$$

For the sake of clarity w.r.t. the subsequent treatment in a discrete setting, we introduce a compact notation of the differential-algebraic system in (10.22). By collecting all geometrical degrees of freedom in the global configuration vector $\boldsymbol{q} = \{\boldsymbol{\varphi}, \operatorname{vec}(\boldsymbol{R}), \phi\} \in \mathbb{R}^n$, the dynamical system yields

$$\boldsymbol{g} = \begin{bmatrix} \dot{\boldsymbol{q}} - \boldsymbol{v} \\ \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{v}} + \boldsymbol{b}(\boldsymbol{v}) + \nabla \boldsymbol{\Phi}^T(\boldsymbol{q}) \boldsymbol{\lambda} + \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u}) \\ \boldsymbol{\Phi}(\boldsymbol{q}) \end{bmatrix} = \boldsymbol{0} \tag{10.23}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the configuration dependent global mass matrix,

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} M_{\varphi}^* \boldsymbol{I}_3 & \boldsymbol{s}^{*T} \otimes \boldsymbol{I}_3 & \boldsymbol{0}_{(3,1)} \\ \boldsymbol{s}^* \otimes \boldsymbol{I}_3 & \boldsymbol{E}^* \otimes \boldsymbol{I}_3 & (\boldsymbol{e}_i \otimes \mathcal{E} \boldsymbol{R} \hat{\boldsymbol{C}}_1) \boldsymbol{e}_i \\ \boldsymbol{0}_{(1,3)} & [(\boldsymbol{e}_i \otimes \mathcal{E} \boldsymbol{R} \hat{\boldsymbol{C}}_1) \boldsymbol{e}_i]^T & 2\mathcal{E} \end{bmatrix} \tag{10.24}$$

$\boldsymbol{b}(\boldsymbol{v}) \in \mathbb{R}^n$ denotes the auxiliary coupling terms

$$\boldsymbol{b}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{0}_{(3,1)} \\ (\boldsymbol{e}_i \otimes 2\mathcal{E} \dot{\phi} \dot{\boldsymbol{R}} \hat{\boldsymbol{C}}_1) \boldsymbol{e}_i \\ 0_{(1,1)} \end{bmatrix} \tag{10.25}$$

and $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u}) \in \mathbb{R}^n$ accounts for external forces.

10.4 Optimal Control

We introduce the general augmented cost functional

$$\hat{J} = \boldsymbol{\mu}^f \cdot \boldsymbol{\Psi}(\boldsymbol{x}_N)\Big|_{t_f} + \int_0^T \left[J(\boldsymbol{u}) + \boldsymbol{\mu}^e \cdot \boldsymbol{g}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{u}) \right] \mathrm{d}t \tag{10.26}$$

where $\boldsymbol{\Psi}, \boldsymbol{\mu}^f \in \mathbb{R}^{2n}$ and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$, $\boldsymbol{\mu}^e$ account for the terminal conditions and state equations together with their respective Lagrangian multipliers and costate variables $\boldsymbol{\mu}^\alpha$, $\alpha \in \{f, e\}$. Here, $\boldsymbol{u}$ denotes the control inputs and $\boldsymbol{x} = \{\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\lambda}\}$ comprises the configuration on position and velocity level $\boldsymbol{q}$, $\boldsymbol{v}$ as well as the dynamical Lagrangian multipliers $\boldsymbol{\lambda}$. Note that the subscript notation $\bullet|_{t_f}$ refers to the fact that the terminal constraints $\boldsymbol{\Psi}(\boldsymbol{x}_N)$ apply to the final time node $t_N = t_f$.

For simplicity in the present work, the running cost $J(\boldsymbol{u})$ is assumed to be solely dependent on the control inputs and accounts for the control effort over the respective time domain, i.e.

$$J(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u} \cdot \boldsymbol{u} \tag{10.27}$$

10.4.1 Direct Collocation

In compliance with the policy *"first discretize, then optimize"* for general direct methods, we divide the time domain of interest into smaller intervals

$$t_0 < t_1 < \ldots < t_N = t_f \tag{10.28}$$

where, for the sake of simplicity, all intervals assume the equidistant time step size $h = t_k - t_{k-1}$. Accordingly the discretization of the augmented cost functional (10.26) yields

$$\begin{aligned}\hat{J} &= \sum_{k=1}^{N} \int_{t_{k-1}}^{t_k} \hat{J}_k(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{\mu})\, \mathrm{d}t = \sum_{k=1}^{N} \int_{t_{k-1}}^{t_k} \hat{J}_k(\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\lambda}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{v}}, \boldsymbol{u}, \boldsymbol{\mu})\, \mathrm{d}t \\ &\approx \sum_{k=1}^{N} \hat{J}_k^h(\boldsymbol{q}_{k-\frac{1}{2}}, \boldsymbol{v}_{k-\frac{1}{2}}, \boldsymbol{\lambda}_{k-1,k}, \frac{\boldsymbol{q}_k - \boldsymbol{q}_{k-1}}{h}, \frac{\boldsymbol{v}_k - \boldsymbol{v}_{k-1}}{h}, \boldsymbol{u}_{k-1,k}, \boldsymbol{\mu}_k)\, h = \hat{J}^h\end{aligned} \tag{10.29}$$

Note, that the terminal constraints are discrete from the outset and thus have been omitted for the sake of clarity. In the following, use will be made of either the standard midpoint rule or an energy momentum scheme, when appropriate, as the collocation scheme of choice for the state equations $\boldsymbol{g}$, see also [3]. Note, that both the control

inputs $\boldsymbol{u}_{k-1,k} \in \mathbb{R}^c$ and the dynamical Lagrangian multipliers $\boldsymbol{\lambda}_{k-1,k} \in \mathbb{R}^m$ are assumed to be constant over the respective time domain k. Thus the discrete version of the augmented cost functional reads

$$\hat{J}^h = \boldsymbol{\mu}^f \cdot \boldsymbol{\Psi}(\boldsymbol{x}_N)\Big|_f + \sum_{k=1}^{N} \left\{ \frac{1}{2} \boldsymbol{u}_{k-1,k} \cdot \boldsymbol{u}_{k-1,k}\, h + \boldsymbol{\mu}_k^e \cdot \boldsymbol{g}_k^h \right\} \tag{10.30}$$

where the discrete state equations $\boldsymbol{g}_k^h \in \mathbb{R}^{(2n+m)N}$ are the discrete counterparts of (10.23). The assembly of the complete state equations in a simplified notation yields

$$\boldsymbol{g}^h = \mathop{\mathrm{A}}_{k=1}^{N} \begin{bmatrix} \frac{1}{h}\left(\boldsymbol{q}_k - \boldsymbol{q}_{k-1}\right) - \boldsymbol{v}_{k-\frac{1}{2}} \\ \frac{1}{h}\boldsymbol{M}_1\left(\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\right) + \tilde{\boldsymbol{f}}_k(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{u}_{k-1,k}) \\ \boldsymbol{\Phi}(\boldsymbol{q}_k) \end{bmatrix} = \boldsymbol{0} \tag{10.31}$$

with

$$\tilde{\boldsymbol{f}}_k = \frac{1}{h}\boldsymbol{M}_2(\boldsymbol{q}_{k-\frac{1}{2}})\left(\boldsymbol{v}_k - \boldsymbol{v}_{k-1}\right) + \boldsymbol{b}(\boldsymbol{v}_{k-\frac{1}{2}}) + \nabla\boldsymbol{\Phi}(\boldsymbol{q}_{k-\frac{1}{2}})^T\boldsymbol{\lambda}_{k-1,k} + \boldsymbol{f}(\boldsymbol{q}_{k-1,k}) \tag{10.32}$$

Here, $\boldsymbol{M}_\beta,\ \beta = \{1, 2\}$ denotes partitioned matrices of the global mass matrix (10.24) with either solely constant or configuration dependent inertial entries.

$$\boldsymbol{M}_1 = \begin{bmatrix} M_\varphi^* \boldsymbol{I}_3 & \boldsymbol{s}^{*T} \otimes \boldsymbol{I}_3 & \boldsymbol{0}_{(3,1)} \\ \boldsymbol{s}^* \otimes \boldsymbol{I}_3 & \boldsymbol{E}^* \otimes \boldsymbol{I}_3 & \boldsymbol{0}_{(3,1)} \\ \boldsymbol{0}_{(1,3)} & \boldsymbol{0}_{(1,3)} & 2\mathcal{E} \end{bmatrix} \qquad \boldsymbol{M}_2 = \begin{bmatrix} \boldsymbol{0}_{(3,3)} & \boldsymbol{0}_{(3,9)} & \boldsymbol{0}_{(3,1)} \\ \boldsymbol{0}_{(9,3)} & \boldsymbol{0}_{(9,9)} & (\boldsymbol{e}_i \otimes \mathcal{E}\boldsymbol{R}\hat{\boldsymbol{C}}_1)\boldsymbol{e}_i \\ \boldsymbol{0}_{(1,3)} & [(\boldsymbol{e}_i \otimes \mathcal{E}\boldsymbol{R}\hat{\boldsymbol{C}}_1)\boldsymbol{e}_i]^T & 0 \end{bmatrix} \tag{10.33}$$

10.4.2 Discrete Necessary Conditions of Optimality and Treatment of Terminal Conditions

Next, we elaborate on the assembly of the **D**iscrete **N**ecessary **C**onditions of **O**ptimality (DNCO), together with the treatment of the terminal conditions within the **L**inear **I**ndependent **C**onstraint **Q**ualification **T**est (LICQT), see also [7]. By partitioning the costate variables $\boldsymbol{\mu} = \left\{\boldsymbol{\mu}^{\boldsymbol{q}}, \boldsymbol{\mu}^{\boldsymbol{v}}, \boldsymbol{\mu}^{\boldsymbol{\lambda}}\right\}$ conjugated to the respective state equation in (10.31) and applying the calculus of variation to the discrete augmented cost function, i.e. $\delta\hat{J}^h = 0$, we obtain the DNCO

$$
\begin{aligned}
\forall\, \delta \boldsymbol{q}_k \quad &: \quad \frac{1}{h}(\boldsymbol{\mu}_k^q - \boldsymbol{\mu}_{k+1}^q) + \frac{\partial \tilde{\boldsymbol{f}}_k}{\partial \boldsymbol{q}_k}^T \boldsymbol{\mu}_k^v + \frac{\partial \tilde{\boldsymbol{f}}_{k+1}}{\partial \boldsymbol{q}_k}^T \boldsymbol{\mu}_{k+1}^v + \frac{\partial \boldsymbol{\Phi}(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k}^T \boldsymbol{\mu}_k^\lambda = \boldsymbol{0} \\
\forall\, \delta \boldsymbol{v}_k \quad &: \quad \frac{1}{h}\boldsymbol{M}_1 \left(\boldsymbol{\mu}_k^v - \boldsymbol{\mu}_{k+1}^v\right) + \frac{\partial \tilde{\boldsymbol{f}}_k}{\partial \boldsymbol{v}_k}^T \boldsymbol{\mu}_k^v + \frac{\partial \tilde{\boldsymbol{f}}_{k+1}}{\partial \boldsymbol{v}_k}^T \boldsymbol{\mu}_{k+1}^v - \frac{1}{2}\left(\boldsymbol{\mu}_{k+1}^q + \boldsymbol{\mu}_k^q\right) = \boldsymbol{0} \\
\forall\, \delta \boldsymbol{\lambda}_{k-1,k} \quad &: \quad \frac{\partial \tilde{\boldsymbol{f}}_k}{\partial \boldsymbol{\lambda}_{k-1,k}}^T \boldsymbol{\mu}_k^v + \frac{\partial \tilde{\boldsymbol{f}}_{k+1}}{\partial \boldsymbol{\lambda}_{k-1,k}}^T \boldsymbol{\mu}_{k+1}^v = \boldsymbol{0} \qquad (10.34) \\
\forall\, \delta \boldsymbol{\mu}_k \quad &: \quad \boldsymbol{g}_k^h(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) = \boldsymbol{0} \\
\forall\, \delta \boldsymbol{\mu}^f \quad &: \quad \boldsymbol{\Psi}(\boldsymbol{x}_N) = \boldsymbol{0} \\
\forall\, \delta \boldsymbol{u}_{k-1,k} \quad &: \quad \frac{\partial J(\boldsymbol{u}_{k-1,k})}{\partial \boldsymbol{u}_{k-1,k}} + \frac{\partial \tilde{\boldsymbol{f}}_k}{\partial \boldsymbol{u}_{k-1,k}}^T \boldsymbol{\mu}_k^v = \boldsymbol{0}
\end{aligned}
$$

Here, $(10.34)_{1-3}$ contain the discrete versions of the costate equations, $(10.34)_4$ the state equations, $(10.34)_5$ the terminal conditions and $(10.34)_6$ the control equations of the optimal control task. For the sake of a compact notation we will treat the DNCO as a general **N**on**l**inear **P**rogramming (NLP) problem, given by

$$
\begin{aligned}
\nabla_{\boldsymbol{y}} J^h(\boldsymbol{u}) + \nabla_{\boldsymbol{y}} \boldsymbol{c}^h(\boldsymbol{y})^T \boldsymbol{\mu} &= \boldsymbol{0} \\
\boldsymbol{c}^h(\boldsymbol{y}) &= \boldsymbol{0}
\end{aligned}
\qquad (10.35)
$$

Here, all variables represent a finite set of unknowns, in particular the global vector $\boldsymbol{y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k, \ldots, \boldsymbol{y}_N\}$ with $\boldsymbol{y}_k = \left\{\boldsymbol{q}_k, \boldsymbol{v}_k, \boldsymbol{\lambda}_{k-1,k}, \boldsymbol{u}_{k-1,k}\right\}$ comprises all unknowns of the optimal control problem except the costate variables $\boldsymbol{\mu}$ in a time node-wise order. Accordingly, the general constraints $\boldsymbol{c}^h(\boldsymbol{y}) = \left\{\boldsymbol{g}^h(\boldsymbol{y}), \boldsymbol{\Psi}(\boldsymbol{x}_N)\right\}$ account for the discrete state equations and terminal conditions. Thus we can write for each time node

$$
\begin{aligned}
\delta \boldsymbol{y}_k \cdot \left[\nabla_{\boldsymbol{y}_k} J(\boldsymbol{u}_{k-1,k}) + \nabla_{\boldsymbol{y}_k} \boldsymbol{g}_k^h(\boldsymbol{y}_{k-1}, \boldsymbol{y}_k)^T \boldsymbol{\mu}_k + \nabla_{\boldsymbol{y}_k} \boldsymbol{g}_{k+1}^h(\boldsymbol{y}_k, \boldsymbol{y}_{k+1})^T \boldsymbol{\mu}_{k+1}\right] &= \boldsymbol{0} \\
\delta \boldsymbol{\mu}_k \cdot \boldsymbol{g}_k^h(\boldsymbol{y}_{k-1}, \boldsymbol{y}_k) &= \boldsymbol{0}
\end{aligned}
\qquad \forall\, \delta \boldsymbol{y}_k \setminus \left\{\delta \boldsymbol{y}_0, \delta \boldsymbol{y}_N\right\} \qquad (10.36)
$$

where we treat the initial state of the optimal control problem as a Dirichlet boundary, i.e. $\delta \boldsymbol{y}_0 = \boldsymbol{0}$. For the LICQT to hold, the assembly of the equations conjugated to the last time nodes assumes the form

$$
\begin{aligned}
\begin{bmatrix} \delta \boldsymbol{y}_{N-1} \\ \delta \tilde{\boldsymbol{y}}_N \end{bmatrix} \cdot \begin{bmatrix} \nabla_{\boldsymbol{y}_{N-1}} J(\boldsymbol{u}_{N-2,N-1}) + \nabla_{\boldsymbol{y}_{N-1}} \boldsymbol{g}_{N-1}^h(\boldsymbol{y}_{N-2}, \boldsymbol{y}_{N-1})^T \boldsymbol{\mu}_{N-1} + \nabla_{\boldsymbol{y}_{N-1}} \tilde{\boldsymbol{g}}_N^h(\boldsymbol{y}_{N-1}, \boldsymbol{y}_N)^T \tilde{\boldsymbol{\mu}}_N \\ \nabla_{\tilde{\boldsymbol{y}}_N} J(\boldsymbol{u}_{N-1,N}) + \nabla_{\tilde{\boldsymbol{y}}_N} \tilde{\boldsymbol{g}}_N^h(\boldsymbol{y}_{N-1}, \boldsymbol{y}_N)^T \tilde{\boldsymbol{\mu}}_N + \nabla_{\tilde{\boldsymbol{y}}_N} \boldsymbol{\Psi}(\boldsymbol{x}_N)^T \boldsymbol{\mu}^f \end{bmatrix} &= \boldsymbol{0} \\
\delta \tilde{\boldsymbol{\mu}}_N \cdot \tilde{\boldsymbol{g}}_N^h(\boldsymbol{y}_{N-1}, \boldsymbol{y}_N)\Big|_N &= \boldsymbol{0} \\
\delta \boldsymbol{\mu}^f \cdot \boldsymbol{\Psi}(\boldsymbol{x}_N)\Big|_f &= \boldsymbol{0}
\end{aligned}
\qquad (10.37)
$$

Here, $\tilde{\boldsymbol{g}}_N^h$ are a reduced set of discrete state equations, see (10.31) with $\boldsymbol{\Phi}(\boldsymbol{q}_N) \rightarrow \tilde{\boldsymbol{\Phi}}(\boldsymbol{q}_N)$, i.e. the constraints $\boldsymbol{\Phi}(\boldsymbol{x}_N)$, that are linear dependent to the terminal conditions $\boldsymbol{\Psi}(\boldsymbol{x}_N)$, must be discarded. Note, that we assume the constraints $\boldsymbol{\Phi}$ on the level of the underlying differential algebraic equations to be linear independent from the outset.

10.5 Numerical Examples

The solution of the DNCO in (10.35) for the respective subsequent optimal control problem has been calculated by IPOPT, a software library for large scale nonlinear optimization, based on an interior point method, see also [13].

10.5.1 Spacecraft Maneuver

First, we will deal with the example of a spacecraft, performing rotational maneuvers by actuating the three-axis mounted reaction wheels. A similar numerical example has been dealt with in [2, 9]. The initial setup of the spacecraft is depicted in (Fig. 10.2), here the director triad of the base body aligns with the spatially fixed inertial frame, i.e. ${}^I\boldsymbol{d}_i = \boldsymbol{e}_i$, together with geometric and inertial parameters, summarized in (Table 10.1). By implementing the rotors of the spacecraft via the proposed gyrostatic formulation, we employ a total set of $n = 15$ non-independent coordinates to describe the configuration of the multibody system at hand, given by

$$\boldsymbol{q} = \left\{ {}^I\boldsymbol{\varphi}, {}^I\boldsymbol{d}_i, {}^{II}\phi, {}^{III}\phi, {}^{IV}\phi \right\} \tag{10.38}$$

Here, solely the orthormality of the base frame has to be enforced by $m = 6$ additional constraints. Concerning the optimal controlled rest-to-rest maneuver, where the goal is to minimize the overall control effort, we state the terminal conditions at time $t_f = 5$ as

$$\Psi = \begin{bmatrix} {}^I\boldsymbol{d}_{i,N} - {}^I\boldsymbol{d}_i^f \\ {}^I\boldsymbol{\omega}_N - {}^I\boldsymbol{\omega}^f \end{bmatrix} \tag{10.39}$$

where the terminal orientation of the base yields ${}^I\boldsymbol{R}^f = \exp \hat{\boldsymbol{b}}\xi \in \{\mathcal{SO}(3) \,|\, {}^I\boldsymbol{R}^f = {}^I\boldsymbol{d}_i^f \otimes \boldsymbol{e}_i\}$ with the axis of rotation and rotation angle for the end configuration, given by $\boldsymbol{b} = \frac{2}{3}\{1, 1, 0.5\}$ and $\xi = \frac{8}{9}\pi$. Note, that a reduced set of terminal constraints on velocity level has been applied, where the connection between the angular velocity and the conjugated director velocities is given by the following formula

$${}^I\boldsymbol{\omega}_\alpha = \frac{1}{2}\boldsymbol{d}_{i,\alpha} \times \dot{\boldsymbol{d}}_{i,\alpha} \qquad \text{for} \qquad \alpha = \{N, f\} \tag{10.40}$$

Naturally the terminal angular velocity ${}^I\boldsymbol{\omega}^f$ is zero from the outset due to the rest-to-rest maneuver.

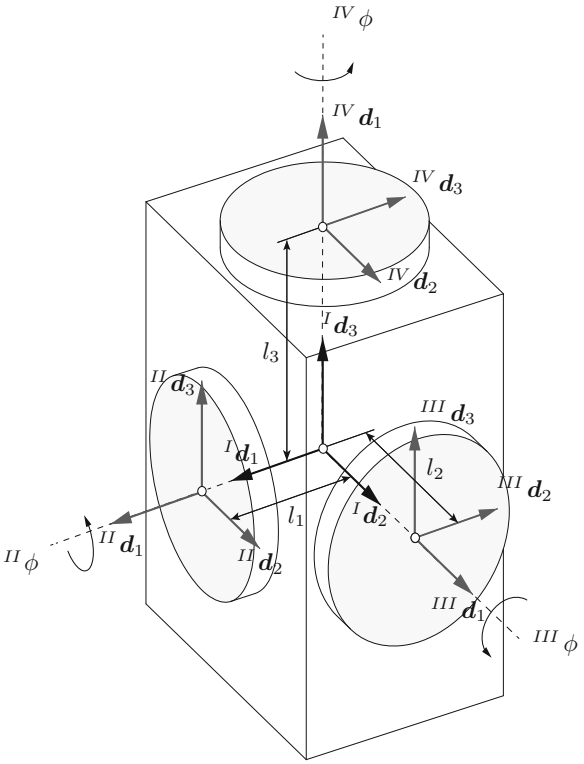


Fig. 10.2 Spacecraft—initial configuration

Table 10.1 Inertial and geometric parameters (Spacecraft)

Body α	I [Base]	II, III, IV [Wheels]
Mass $[{}^{\alpha}M_{\varphi}]$	10	1
Static moment $[{}^{\alpha}\boldsymbol{s}]$	0	0
Euler tensor $[{}^{\alpha}\boldsymbol{E}]$	diag (2.223, 4.408, 7.334)	diag (0.003, 0.141, 0.141)
Length $[l_1, l_2, l_3]$	(0.9167, 1.25, 1.5833)	

Concerning the results, necessary control inputs conjugated to the angular coordinates ${}^{\alpha}\phi$, $\alpha = \{II, III, IV\}$ are depicted in (Fig. 10.3), together with the plot of kinetic, input and total energy in (Fig. 10.4). The actual trajectory of the spacecraft is depicted in (Fig. 10.5), where the magnitude of the respective control inputs is highlighted. Note, that the time domain $\mathcal{T} = [0, 5]$ has been segmented into $N = 160$ equidistant intervals.

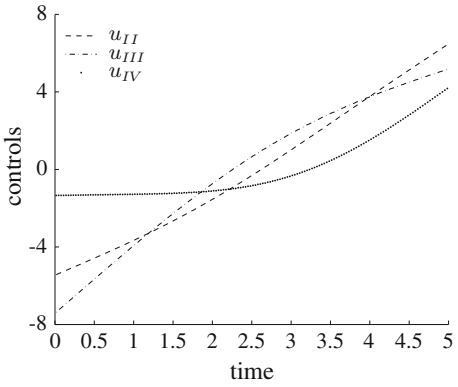


Fig. 10.3 Spacecraft—control inputs

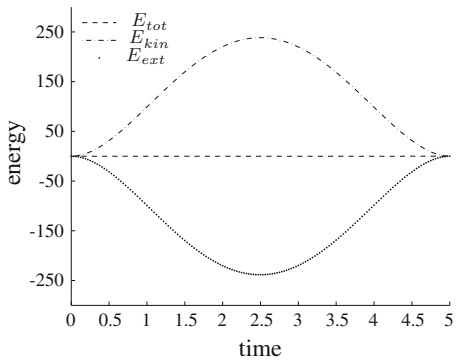


Fig. 10.4 Spacecraft—resulting energies

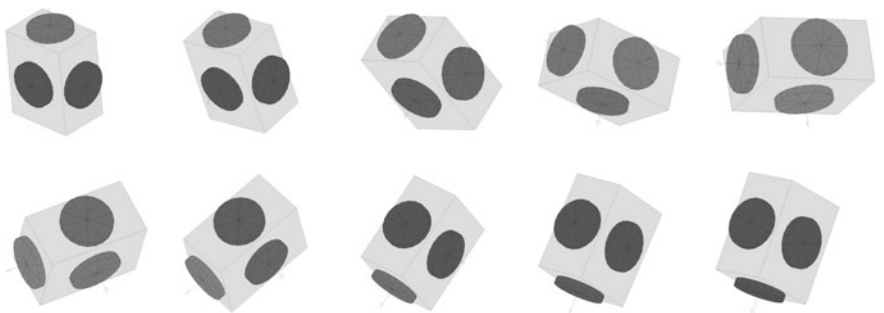


Fig. 10.5 Spacecraft—configuration at $t = \{0.5, 1, \ldots, 4.5, 5\}$

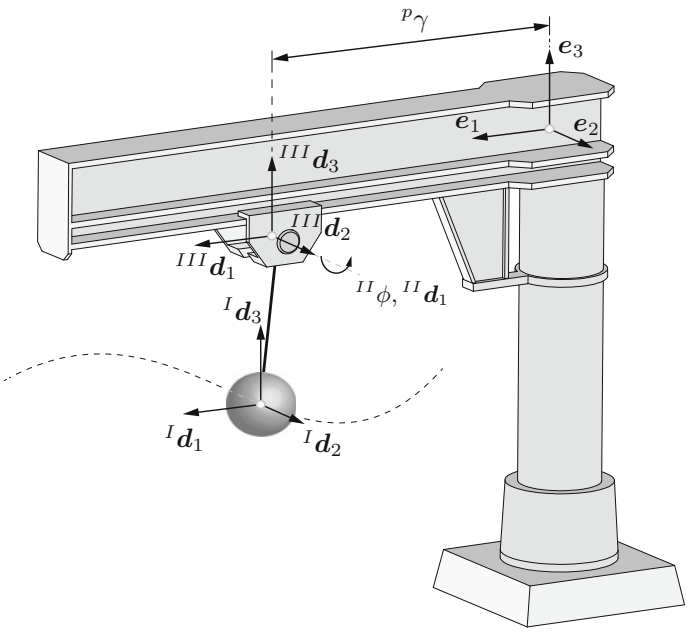


Fig. 10.6 Overhead crane—setup

Table 10.2 Inertial and geometric parameters (Overhead crane)

Body α	I [Trolley]	II [Winch]	III [Mass]
Mass $[{}^{\alpha}M_{\varphi}]$	100	1	9
Static moment $[{}^{\alpha}\boldsymbol{s}]$	0	0	0
Euler tensor $[{}^{\alpha}\boldsymbol{E}]$	diag (1, 0.05, 1)	diag (1, 0.05, 1)	diag (1, 0.05, 1)
Winch radius R	0.1		
Initial rope length	4		

10.5.2 Overhead Crane

As a second numerical example, we will deal with the mechanical system of an overhead crane, see also [11]. The overall topology of the system is depicted in (Fig. 10.6), together with inertial and geometric parameters given in (Table 10.2). Here in particular, the modelling of joints is subject to both the coordinate augmentation technique, see also [12], and the proposed gyrostatic formulation. Thus, the kinematic chain can be described as follows: The spatially fixed cantilever and the trolley (I) interact via an augmented prismatic joint, in this regard, the relative displacement between both bodies is given by ${}^{p}\gamma$. The trolley (I) in turn provides its

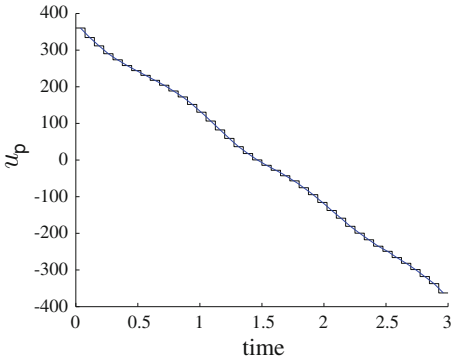


Fig. 10.7 Linear force —augmented prismatic joint

director triad as a subset of coordinates to describe the configuration of the winch (II) within the proposed gyrostatic formulation. Eventually the unrolling of the rope is subject to an augmented distance constraint between the winch (II) and the mass (III), where we make use of the coordinates ${}^{II}\phi$ and ${}^{l}\gamma$, respectively the winch rotational angle and unrolled rope length. Thus we employ a total set of $n = 27$ non-independent coordinates, given by

$$\boldsymbol{q} = \left\{ {}^{I}\boldsymbol{\varphi}, {}^{I}\boldsymbol{d}_i, {}^{II}\phi, {}^{III}\boldsymbol{\varphi}, {}^{III}\boldsymbol{d}_i, {}^{l}\gamma, {}^{p}\gamma \right\} \tag{10.41}$$

Since the trolley is restricted to a movement along the cantilever due to the prismatic joint, together with the linear dependancy between the winch rotation angle and unrolled rope length, the rest-to-rest maneuver is defined by the initial and final configuration $\boldsymbol{q}_0 = \left\{ {}^{III}\boldsymbol{\varphi}_0, {}^{p}\gamma_0 \right\} = \{0, 0, -4, 0\}$ and $\boldsymbol{q}^f = \left\{ {}^{III}\boldsymbol{\varphi}^f, {}^{p}\gamma^f \right\} = \{5, 0, -1, 5\}$. Eventually, we can state the terminal conditions at time $t_f = 3$

$$\boldsymbol{\Psi} = \begin{bmatrix} {}^{III}\boldsymbol{\varphi}_N - {}^{III}\boldsymbol{\varphi}^f \\ {}^{p}\gamma - {}^{p}\gamma^f \\ {}^{III}\dot{\boldsymbol{\varphi}}_N - {}^{III}\dot{\boldsymbol{\varphi}}^f \\ {}^{p}\dot{\gamma} - {}^{p}\dot{\gamma}^f \end{bmatrix} \tag{10.42}$$

As before, the conjugated terminal velocities are enforced to be zero in compliance with a rest-to-rest trajectory. Concerning the numerical results, the control input u_p, conjugated to the augmented prismatic joint, and the control input u_ϕ, conjugated to the relative twist of the winch, are depicted in (Figs. 10.7 and 10.8). For Snapshots of the optimal rest-to-rest maneuver see (Fig. 10.9). Note, that the time domain $\mathcal{T} = [0, 3]$ has been segmented into $N = 80$ equidistant intervals for the present example.

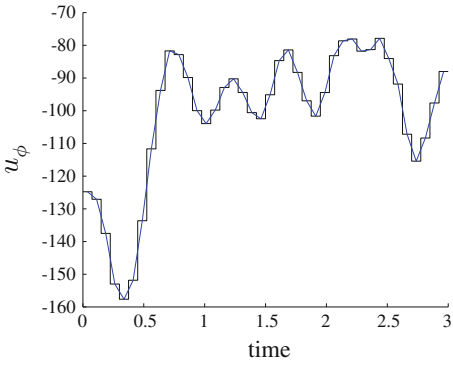


Fig. 10.8 Torque—winch (slave)

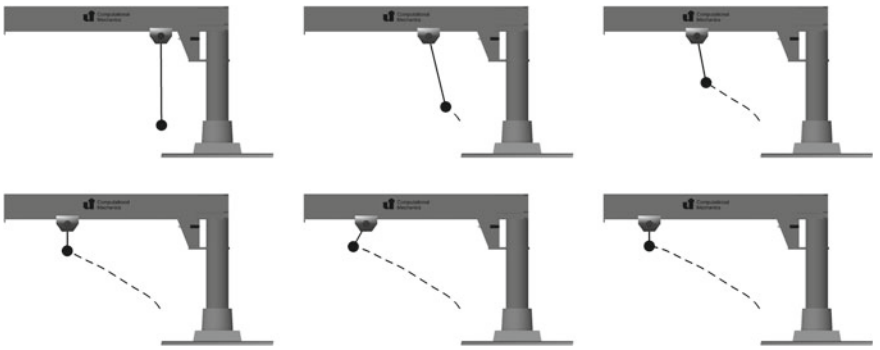


Fig. 10.9 Overhead crane—configuration at $t = \{0, 0.75, 1.5, 2.25, 2.625, 3\}$ ($N = 80$)

10.6 Conclusion

In the present work a new formulation based on natural coordinates for rigid bodies with inertial symmetry, i.e. rotors, has been proposed. The newly developed formulation is used as control input interface for optimal control problems. The equations of motion pertaining to the new formulation assume the form of DAEs. After the discretization in time the DAEs give rise to discrete state equations being used as constraints in a parameter optimization problem. Thus a direct transcription approach is applied to solve the optimal control problem. The newly proposed rigid body formulation is especially-well suited for the description of spinning bodies such as momentum wheels and winches. Two numerical examples have been presented to demonstrate the capabilities of the proposed formulation.

Acknowledgments Support for this research was provided by the Deutsche Forschungsgemeinschaft (DFG) under Grant BE 2285/10-1. This support is gratefully acknowledged.

References

1. Betsch P, Sänger N (2012) On the consistent formulation of torques in a rotationless framework for multibody dynamics. Computers and structures, http://dx.doi.org/10.1016/j.compstruc.2012.10.005
2. Betsch P, Siebert R, Sänger N (2012) Natural coordinates in the optimal control of multibody systems. J Comput Nonlinear Dynam 7(1):011009/1–8
3. Betsch P, Steinmann P (2001) Constrained integration of rigid body dynamics. Comput Methods Appl Mech Engrg 191:467–488
4. Betsch P, Uhlar S (2007) Energy-momentum conserving integration of multibody dynamics. Multibody Syst Dyn 17(4):243–289
5. Bottasso CL, Croce A (2004) Optimal control of multibody systems using an energy preserving direct transcription method. Multibody Syst Dyn 12(1):17–45
6. Greenwood DT (1988) Principles of dynamics. Prentice-Hall, Upper Saddle River
7. Gros S, Zanon M, Vukov M, Diehl M (2012) Nonlinear MPC and MHE for mechanical multibody systems with application to fast tethered airplanes. In: Proceedings of 4th IFAC nonlinear model predictive control conference, Noordwijkerhout, NL, August 2012, pp 23–27
8. Koch MW, Leyendecker S (2012) Energy momentum consistent force formulation for the optimal control of multibody systems. Multibody Sys Dyn, http://dx.doi.org/10.1007/s11044-012-9332-9
9. Leyendecker S, Ober-Blöbaum S, Marsden JE, Ortiz M (2010) Discrete mechanics and optimal control for constrained systems. Optimal Control Appl Methods 31(6):505–528
10. Moon FC (2008) Applied dynamics with applications to multibody and mechatronic systems, 2nd edn. WILEY-VCH, Weinheim
11. Siebert R (2012) Mechanical integrators for the optimal control in multibody dynamics. Phd thesis, University of Siegen
12. Uhlar S, Betsch P (June 2007) On the rotationless formulation of multibody dynamics and its conserving numerical integration. In: Bottasso CL, Masarati P, Trainelli L (eds) Proceedings of the ECCOMAS thematic conference on multibody dynamics. Italy, Milano, pp 25–28
13. Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Prog 106:25–57

Chapter 11
Development of Tether Space Mobility Device

Shoichiro Takehara, Takahiro Nishizawa, Masaya Kawarada, Kazunori Hase and Yoshiaki Terumichi

Abstract With the increasing use of the International Space Station, humans have more opportunities to work in space. In space, a mobility device that operates efficiently is needed. In this research, a mobility system called the "Tether Space Mobility Device" (hereinafter called TSMD) is proposed. In general, the tether is a cable or a wire rope. The proposed system has a mechanism that uses the tether for enabling a human to move to a target point. However, this system has the problem that the center of mass of the human and that of the TSMD are different from the direct line to the target point. Then, the human is rotated by the tension of the tether. Thus, to use this device safely, rotation of the human body must be controlled. For this reason, a numerical simulation model is proposed. The numerical model is composed of three rigid bodies and one flexible body that can express motion with large deformation and large displacement. In this model, winding motion of the tether can be expressed. An experiment of the TSMD was designed to move under two-dimensional microgravity. The experiment confirmed the validity of the numerical simulation model. The possibility of the mobility device using the tether and the influence of the control system are discussed.

S. Takehara (✉) · Y. Terumichi
Department of Engineering and Applied Sciences, Faculty of Science and Technology, Sophia University, 7-1 Kioi-cho, Chiyoda-ku, Tokyo, Japan
e-mail: stakeha@sophia.ac.jp

Y. Terumichi
e-mail: y-terumi@sophia.ac.jp

T. Nishizawa · M. Kawarada · K. Hase
Department of Mechanical Engineering, Tokyo Metropolitan University, 1-1 Minami-Ohsawa, Hachioji, Tokyo, Japan
e-mail: kazunori.hase@tmu.ac.jp

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_11,

11.1 Introduction

With the increasing use of the International Space Station, humans have more opportunities to work in space. In this situation, a mobility device that works efficiently is needed. But some problem must be solved. First, the human body is suspended without the force of gravity. Second, the air cannot be polluted in the closed space of the Space Station. Thus, an air-polluting mobility device such as a device using a gas-fueled thruster is unacceptable. In this research, a mobility system called the "Tether Space Mobility Device" (hereinafter called TSMD) is proposed. In general, the tether is a cable or a wire rope. The cable and wire rope have the advantage of being light weight and compact, and the system using a tether does not require a thruster that needs fuel for moving. The system using a tether has various possible applications in extreme environmental conditions, such as in space. The tethers are expected to shift orbits and to move robots in space [1–5]. As an example of the interest in tethered systems, a tether rocket experiment was conducted on a sounding rocket launched in 2010 [6]. For the abovementioned applications, practical experimentation with the actual system or a representative full-scale apparatus is not practicable. Consequently, accurate models that allow numerical simulation are very important.

In tethered systems, the tension of the tether has a large influence on the motion of the system, even if the tension is small, because the gravitational force that acts in a constant direction in space has little influence on the motion. In such circumstances, the motion of the tether often has large rotation and deformation. Moreover, it is assumed that the motion of the tethered system is complex because the coupling motion between the tether and the attached equipment is excited. Therefore, the tethered system should be modeled as a flexible body attached to rigid bodies. In addition it is important for the proposed system to consider the motion of winding the tether. In this research, the TSMD model is composed of three rigid bodies and one flexible body that can express motion with large deformation and large displacement. The flexible body is formulated by absolute nodal coordinate formulation. Winding motion is also considered in this model. The model that retracts the tether in the machine is formulated. The tether is wound by the arm contacting the tether and the rigid body. The interaction between the TSMD and the tether is investigated.

11.2 Concept of the TSMD

The basic concept of the TSMD is shown in Fig. 11.1. This system has a mechanism that uses the tether for moving. The tether is shot out of the TSMD and the end of the tether is attached to the target point. Then, the tether is wound by a motor and the human moves toward the target point. However, this system has the problem that the center of mass of the human and that of the TSMD are different from the direct line between the target point and the part held by the human. In this case, the human

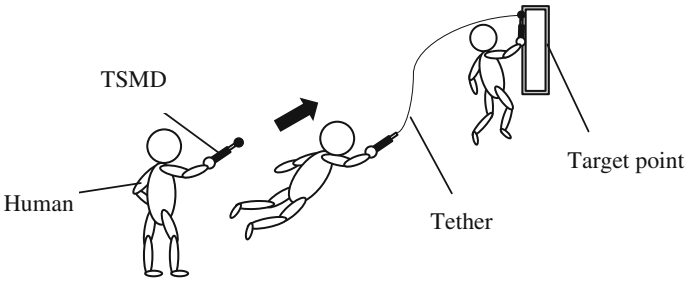
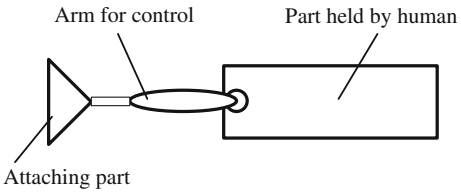


Fig. 11.1 Concept of the TSMD

Fig. 11.2 Composition of the TSMD



is rotated by the tension of the tether. The coupling motion between the tether and the rigid bodies is excited [7]. Therefore, the direction of tension and the velocity of winding need to be controlled. In Fig. 11.2, an outline of the TSMD is shown. The TSMD is composed of the part attaching to the target, an arm for controlling the tether, and the part held by the human.

11.3 Modeling and Formulation of the TSMD

11.3.1 Analytical Model

Figure 11.3 shows the analytical model of the TSMD. The TSMD model is composed of three rigid bodies and a flexible body, which expresses motion with large deformation and large displacement. The flexible body is formulated by absolute nodal coordinate formulation (ANCF) [8–10]. In this research, the center of gravity and the moment of inertia about the human body are considered. The influence of the motion of the human body is intended to be a future task. That is to say, the angle of rigid body 2 is locked in the angle of rigid body 3. The main types to treat the motion of winding are to change length of the elements and to move from unconstraint area into constraint area [11–14]. In this system later method is adapted because the contact phenomenon between the arm and the tether is important. In this model, the tether is wound by the contacting arm. Figure 11.4 shows the modeling of the winding tether. The interaction between the rigid bodies (TSMD and human) and the flexible

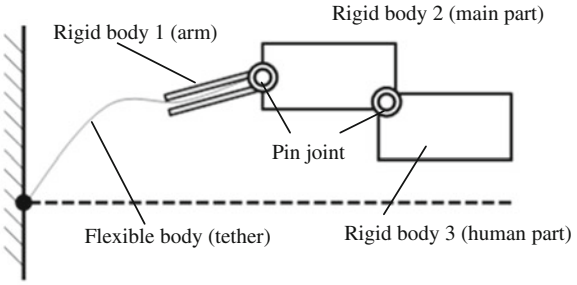


Fig. 11.3 Modeling of the TSMD

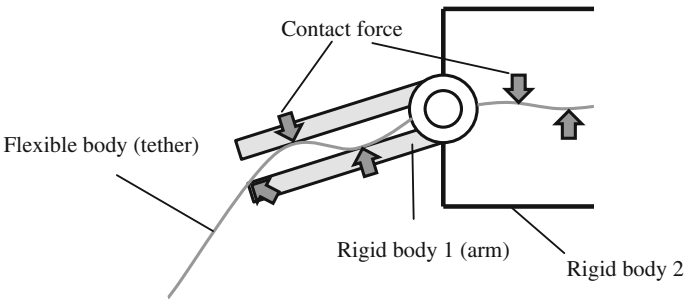


Fig. 11.4 Modeling of the winding

body (tether) can be investigated in this model. The interaction occurs according to the reaction force and the tether tension. The reaction force is calculated when the flexible body (tether) contacts the edge and the inner surface of rigid body 1 (arm). Because the tether is assumed to pass through the inside of rigid body 1, this model distinguishes the node that contacts rigid body 1. Then, a suitable reaction force can be defined.

11.3.2 Formulation of the System

In this section the motion of the flexible body is formulated using ANCF which was proposed by Shabana et al. [9, 10]. In this formulation, it is easy to describe the motion of the flexible body with large deformation, rotation and translation displacement. Global coordinates and position vector gradients are used as the nodal coordinates.

First, this formulation is extended to flexible body motion with rigid bodies.We divide set the tether into n pieces by ANCF. The motion equation of the tether is

$$\mathbf{M}_t \ddot{\mathbf{e}}_t = \mathbf{Q}_t \tag{11.1}$$

where $\mathbf{M}_t$ is the mass matrix, and $\mathbf{e}_t \equiv \begin{bmatrix} e_1 & e_2 & \dots & e_{4(n+1)} \end{bmatrix}^T$ is the absolute nodal coordinate, where e_{4i+1} and e_{4i+2} are the position of each nodes, e_{4i+3} and e_{4i+4} are the spatial derivatives of each nodes. The coordinates e_{4n+1} and e_{4n+2} are the connecting position between flexible body and rigid bodies, e_{4n+3} and e_{4n+4} are the spatial derivatives of the displacements of the connecting position between flexible body and rigid bodies ($i = 0, 1, \dots, n-1, n$). In this formulation, the following shape function is used.

$$\mathbf{S} = \begin{bmatrix} 1-3\xi^2+2\xi^3 & 0 \\ 0 & 1-3\xi^2+2\xi^3 \\ l_e(\xi-2\xi^2+\xi^3) & 0 \\ 0 & l_e(\xi-2\xi^2+\xi^3) \\ 3\xi^2-2\xi^3 & 0 \\ 0 & 3\xi^2-2\xi^3 \\ l_e(\xi^3-\xi^2) & 0 \\ 0 & l_e(\xi^3-\xi^2) \end{bmatrix}^T \tag{11.2}$$

where $\xi = x/l_e$, x is the coordinate along the body axis in the initial configuration and l_e is the length of the element.

$\mathbf{Q}_t \equiv \begin{bmatrix} Q_1 & Q_2 & \dots & Q_{4(n+1)} \end{bmatrix}^T$ consists of the external force and the elastic force. In this model, the authors suppose that the longitudinal deformation of an element is minimal when an element does not rotate at high angular velocity. Thus, geometrically-approximated elastic force is used [15, 16].

The motion of the equation of the rigid bodies is

$$\mathbf{M}_r \ddot{\mathbf{q}}_r = \mathbf{Q}_r \tag{11.3}$$

where $\mathbf{M}_r = diag\begin{bmatrix} m_{r1} & m_{r1} & I_{r1} & m_{r2} & m_{r2} & I_{r2} & m_{r3} & m_{r3} & I_{r3} \end{bmatrix}^T$ is the mass matrix, m_{ri} is the mass of body m, and I_{ri} is the inertia moment of the rigid bodies, ($i = 1, 2, 3$). Moreover, $\mathbf{q}_r = \begin{bmatrix} x_{r1} & y_{r1} & \theta_{r1} & x_{r2} & y_{r2} & \theta_{r2} & x_{r3} & y_{r3} & \theta_{r3} \end{bmatrix}^T$ is the general coordinates of the rigid bodies. The coordinates x_{ri} and y_{ri} are the center of body m_{ri}, θ_{ri} is the rotation of the rigid bodies ($i = 1, 2, 3$).

$\mathbf{Q}_r = \begin{bmatrix} F_{r1} & F_{r1} & M_1 & F_{r2} & F_{r2} & M_2 & F_{r3} & F_{r3} & M_3 \end{bmatrix}^T$ consists of the external force and the moment of the rigid bodies.

The motion equation of the system is

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q} \tag{11.4}$$

where $\mathbf{q} = \begin{bmatrix} \mathbf{e}_t^T & \mathbf{q}_r^T \end{bmatrix}^T, \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_t^T & \mathbf{Q}_r^T \end{bmatrix}^T, \mathbf{M} = \begin{bmatrix} \mathbf{M}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_r \end{bmatrix}$. The differential algebraic equations are obtained as

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^{\mathbf{T}} \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A \\ \boldsymbol{\gamma} \end{bmatrix} \tag{11.5}$$

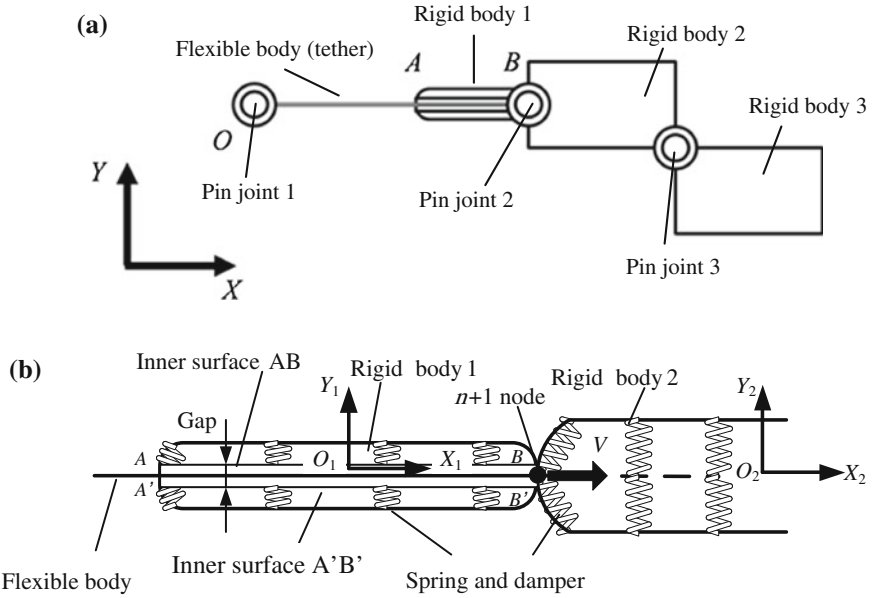


Fig. 11.5 Constraint of the system. **a** Overall view of the constraint. **b** The detail inside of the rigid bodies

where $\Phi_{\mathbf{q}}$ is the Jacobean matrix, $\boldsymbol{\lambda}$ is Lagrange multipliers, Differentiating the constraint twice with respect to time, $\Phi_{\mathbf{q}}\ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\Phi_{\mathbf{qt}}\dot{\mathbf{q}} - \Phi_{\mathbf{tt}} \equiv \boldsymbol{\gamma}$.

Next, we formulate the constraint of the system. Figure 11.5 shows the modeling of the constraint. As shown in Fig. 11.5a, the flexible body is connected with point O by pin joint 1. Rigid body 1 is connected to the rigid body 2 by pin joint 2. Rigid body 2 is connected to rigid body 3 by pin joint 3. In this report, the angle of rigid body 2 is locked by the angle of rigid body 3. Furthermore, the nodes of the end of the flexible body are set at the connecting point between rigid bodies 1 and 2 as an initial condition. As shown in Fig. 11.5b, the local coordinates of rigid bodies 1 and 2 are defined as O_1–X_1Y_1 and O_2–X_2Y_2 respectively. Then, the end of the flexible body moves in the X_2 direction at speed V when the flexible body is wound. When the node of the flexible body contacts the inner surface AB, A′B′, and the edge of the rigid body 1, the contact force is caused by a spring-damper. The gap is set in rigid body 1. The flexible body in rigid body 2 is fixed by a hard spring on the X_2 axis.

The 1st node of the flexible body is connected to the origin of the absolute coordinate system by pin joint 1. Rigid body 1 and rigid body 2 are connected. Rigid body 2 and rigid body 3 are connected. Then, the constraint of the system is described as Eqs. (11.6), (11.7) and (11.8).

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \mathbf{0} \tag{11.6}$$

$$\begin{bmatrix} x_{r2} \\ y_{r2} \end{bmatrix} + \mathbf{T}_2 \begin{bmatrix} -\frac{l_{r2}}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} x_{r1} \\ y_{r1} \end{bmatrix} - \mathbf{T}_1 \begin{bmatrix} \frac{l_{r1}}{2} \\ 0 \end{bmatrix} = \mathbf{0} \tag{11.7}$$

$$\begin{bmatrix} x_{r3} \\ y_{r3} \end{bmatrix} + \mathbf{T}_3 \begin{bmatrix} -\frac{l_{r3}}{2} \\ \frac{h_{r3}}{2} \end{bmatrix} - \begin{bmatrix} x_{r2} \\ y_{r2} \end{bmatrix} - \mathbf{T}_2 \begin{bmatrix} \frac{l_{r2}}{2} \\ -\frac{h_{r3}}{2} \end{bmatrix} = \mathbf{0} \tag{11.8}$$

where l_{ri}, h_{ri} $(i = 1, 2, 3)$ are the length and width of the ith rigid body, respectively. $\mathbf{T}_i$ $(i = 1, 2, 3)$ is transform matrices from local coordinate system to absolute coordinate system. Then, the constraint considered is when the last node of the flexible body moves in the X_2 direction of the local coordinates of rigid body 2 at speed V. When the position vector of the n+1th node of the flexible body is expressed by vector $\mathbf{d}$ in the local coordinates of rigid body 2, the constraint is described by Eq. (11.9).

$$\mathbf{R}_A - \mathbf{R}_2 - \mathbf{T}_2\mathbf{d} = \mathbf{0} \tag{11.9}$$

where $\mathbf{R}_A = [e_{4n+1} e_{4n+2}]^T$ is the position vector of the n+1th node of the flexible body, $\mathbf{R}_2 = [x_{r2}\ y_{r2}]^T$ is the position vector of the center of rigid body 2. The position vector $\mathbf{d}$ in the local coordinates of rigid body 2, which moves in the X_2 direction at speed V from initial position $\mathbf{d}_0 = [-l_{r2}/2\ 0]^T$, is described as

$$\mathbf{d} = \mathbf{d_0} + \begin{bmatrix} \int_0^t V dt \\ 0 \end{bmatrix} \tag{11.10}$$

Therefore, the constraint equations of the system can be written as

$$\Phi = \begin{bmatrix} e_1 \\ e_2 \\ e_{4n+1} - x_{r2} - \left(-\frac{l_{r2}}{2} + \int_0^t V dt \right) \cos\theta_{r2} \\ e_{4n+2} - y_{r2} - \left(-\frac{l_{r2}}{2} + \int_0^t V dt \right) \sin\theta_{r2} \\ x_{r2} - \frac{l_{r2}}{2}\cos\theta_{r2} - x_{r1} - \frac{l_{r1}}{2}\cos\theta_{r1} \\ y_{r2} - \frac{l_{r2}}{2}\sin\theta_{r2} - y_{r1} - \frac{l_{r1}}{2}\sin\theta_{r1} \\ x_{r3} + \frac{l_{r3}}{2}\cos\theta_{r3} + \frac{h_{r3}}{2}\sin\theta_{r3} - \left(x_{r2} + \frac{l_{r2}}{2}\cos\theta_{r2} + \frac{h_{r2}}{2}\sin\theta_{r2} \right) \\ y_{r3} - \frac{l_{r3}}{2}\sin\theta_{r3} + \frac{h_{r3}}{2}\cos\theta_{r3} - \left(y_{r2} + \frac{l_{r2}}{2}\sin\theta_{r2} - \frac{h_{r2}}{2}\cos\theta_{r2} \right) \end{bmatrix} = \mathbf{0} \tag{11.11}$$

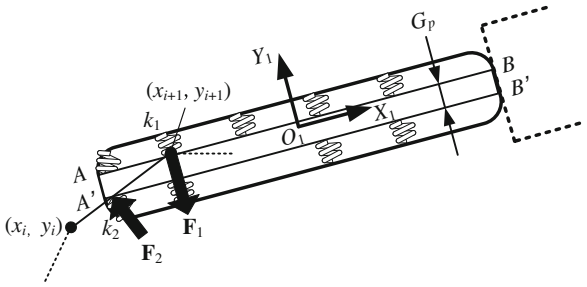


Fig. 11.6 The element of the flexible body is inhaled inrigid body 1

Next, the contact between the flexible body and the rigid body 1 is considered. The contact force between the flexible body and the rigid body is calculated by a spring and a damper element. Figure 11.6 shows the ith flexible body element drawn into rigid body 1. Here, $\mathbf{F}_1$is the reaction force by rigid body 1 in the case that the node of the element is retracted completely. $\mathbf{F}_2$ is the reaction force at the edge of rigid body 1 when the node of the element is being retracted.

First, the case that the i+1th node contacts the inner surface AB of rigid body 1 is considered. In this case, the position vector $[x_{i+1}\, y_{i+1}]^T$ of the i+1th node in absolute coordinates is

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \mathbf{R_1} + \mathbf{T}_1 \begin{bmatrix} \bar{x}^1_{i+1} \\ \bar{y}^1_{i+1} \end{bmatrix} \tag{11.12}$$

where $\mathbf{R_1}$ is the position vector of rigid body 1 in absolute coordinates, and $\begin{bmatrix} \bar{x}^1_{i+1} & \bar{y}^1_{i+1} \end{bmatrix}^T$ is the position vector of rigid body 1 in local coordinates.

$$\begin{bmatrix} \bar{x}^1_i \\ \bar{y}^1_i \end{bmatrix} = \mathbf{T}_1^{-1} \left(\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} - \mathbf{R_1} \right) \tag{11.13}$$

Reaction force $\mathrm{F_{k1}}$ by the spring and damper is described as

$$\left.\begin{aligned} \bar{\mathbf{F}}^1_{k1} &= -k_1 \begin{bmatrix} 0 \\ \bar{y}^1_{i+1} - \mathrm{sgn}(\bar{y}^1_{i+1}) \frac{G_p}{2} \end{bmatrix} - c_1 \begin{bmatrix} 0 \\ \dot{\bar{y}}^1_{i+1} \end{bmatrix} \quad \left(\left| \bar{y}^1_{i+1} \right| \geq \frac{G_p}{2} \right) \\ \bar{\mathbf{F}}^1_{k1} &= 0 \quad \left(\left| \bar{y}^1_{i+1} \right| < \frac{G_p}{2} \right) \end{aligned}\right\} \tag{11.14}$$

where $\bar{\mathbf{F}}^1_{k1}$ shows $\mathbf{F}_{\mathrm{k1}}$ in the local coordinate system, $\mathbf{G}$p is the distance between inner surface AB and A′B′. Frictional force can be written as follows:

$$\bar{\mathbf{F}}^1_{f1} = \begin{bmatrix} -\mu_1 \bar{F}^1_{k1y} \\ 0 \end{bmatrix} \tag{11.15}$$

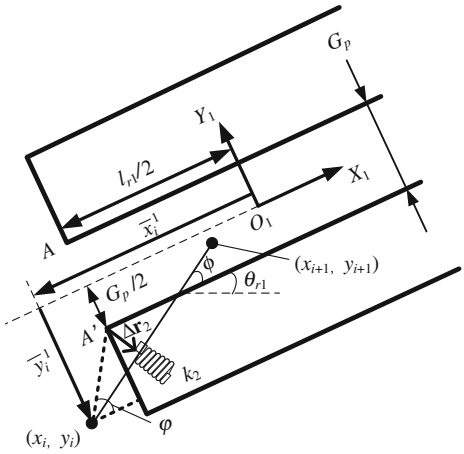


Fig. 11.7 Magnified figure of edge of rigid body 1

where μ_1 is the friction coefficient, $\bar{F}^1_{k1y}$ is the component of the y direction, and $\bar{\mathbf{F}}^1_{f1}$ is $\mathbf{F}_f$ in the local coordinate system. Then,as the reaction force from AB, $\mathbf{F}_1$ is

$$\left.\begin{aligned}\mathbf{F}_1 &= \mathbf{T}_1 \begin{bmatrix} -\mu_1 \left\{ -k_1 \left(\bar{y}^1_{i+1} - \mathrm{sgn}(\bar{y}^1_{i+1}) \frac{G_p}{2} \right) - c_1 \dot{\bar{y}}^1_{i+1} \right\} \\ -k_1 \left(\bar{y}^1_{i+1} - \mathrm{sgn}(\bar{y}^1_{i+1}) \frac{G_p}{2} \right) - c_1 \dot{\bar{y}}^1_{i+1} \end{bmatrix} \quad \left(\left| \bar{y}^1_{i+1} \right| \geq \frac{G_p}{2} \right) \\ \mathbf{F}_1 &= \mathbf{0} \quad \left(\left| \bar{y}^1_{i+1} \right| < \frac{G_p}{2} \right) \end{aligned}\right\} \tag{11.16}$$

The force and moment acting on rigid body 1 are resultant forces of the reaction force of elements in rigid body 1. These forces are described as

$$\begin{bmatrix} \mathbf{F}_{r1} \\ M_{r1} \end{bmatrix} = \begin{bmatrix} \sum (-\mathbf{F}_1) \\ \sum \left\{ \begin{bmatrix} \bar{x}^1_{i+1} \\ \bar{y}^1_{i+1} \end{bmatrix} \times (-\mathbf{F}_1) \right\} \end{bmatrix} \tag{11.17}$$

In a similar way, the reaction force from $A'B'$ and the flexible body is obtained. Substituting the local coordinates of rigid body 1 into those of rigid body 2 and setting Gp = 0, the reaction force acting from rigid body 2 on the flexible body can be obtained. However, it is supposed that rigid body 2 gains no reaction force because the element of the flexible body that is drawn in rigid body 2 is considered to be stored.

Next, reaction force $\mathbf{F}_2$ of the element of the flexible body from the edge of rigid body 1 is set. Figure 11.7 shows the magnified figure of the edge of rigid body 1.

The reaction force at point A' is considered. For simplicity, it is assumed that this flexible body element does not deform geometrically. The reaction force at point A' is described as

$$\mathbf{F}_{k2} = -k_2 \Delta \mathbf{r}_2 - c_2 \Delta \dot{\mathbf{r}}_2 \tag{11.18}$$

Then, the unit vector of $\Delta \mathbf{r}_2$ is defined as

$$\mathbf{i}_r = \frac{1}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \begin{bmatrix} y_{i+1} - y_i \\ -(x_{i+1} - x_i) \end{bmatrix} \tag{11.19}$$

From the geometric relation, the magnitude of $\Delta \mathbf{r}_2$ is described as

$$|\mathbf{r}_2| = \left\{ -\bar{y}_i^1 - \mathrm{sgn}(\bar{y}_{i+1}^1) \left(-\bar{x}_i^1 - \frac{l_{r1}}{2} \right) \tan\phi - \mathrm{sgn}(\bar{y}_{i+1}^1) \frac{G_p}{2} \right\} \cos\phi \tag{11.20}$$

where ϕ is the angle of the ith element on the local coordinates of rigid body 1. Then, ϕ is written as

$$\phi = \left| \arctan \left(\frac{\bar{y}_{i+1}^1 - \bar{y}_i^1}{\bar{x}_{i+1}^1 - \bar{x}_i^1} \right) \right| \tag{11.21}$$

Using Eqs. (11.19) and (11.20), $\Delta \mathbf{r}_2$ is obtained as

$$\Delta \mathbf{r}_2 = \frac{\left\{ -\bar{y}_i^1 - \mathrm{sgn}(\bar{y}_{i+1}^1) \left(-\bar{x}_i^1 - \frac{l_{r1}}{2} \right) \tan\phi - \mathrm{sgn}(\bar{y}_{i+1}^1) \frac{G_p}{2} \right\} \cos\phi}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \begin{bmatrix} y_{i+1} - y_i \\ -(x_{i+1} - x_i) \end{bmatrix} \tag{11.22}$$

Substituting Eq. (11.22) into (11.18), $\mathbf{F}_{k2}$, which is defined by the spring and damping elements, is obtained. Here, considering the direction of retracting, the unit vector of the friction force at point A' is obtained as

$$\mathbf{i}_f = \frac{1}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \begin{bmatrix} -(x_{i+1} - x_i) \\ -(y_{i+1} - y_i) \end{bmatrix} \tag{11.23}$$

In the same way, $\mathbf{F}_{f2}$ is described as

$$\mathbf{F}_{f2} = \frac{\mu_1 |\mathbf{F}_{k2}|}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \begin{bmatrix} -(x_{i+1} - x_i) \\ -(y_{i+1} - y_i) \end{bmatrix} \tag{11.24}$$

Therefore, letting the force that the ith element gains from point A' be $\mathrm{F_2}^{(i)}$, $\mathrm{F_2}^{(i)}$ is written as

$$\left.\begin{aligned} \mathbf{F}_2^{(i)} &= \mathbf{F}_{k2} + \mathbf{F}_{f2} \quad (\phi < \varphi) \\ \mathbf{F}_2^{(i)} &= \mathbf{0} \quad (\phi \geq \varphi) \end{aligned}\right\} \tag{11.25}$$

where φ is defined as the angle between the ith nodal in the local coordinates of rigid body 1 and point A'. Here, φ is as

$$\phi = \left| \arctan\left(\frac{-\mathrm{sgn}(\bar{y}_{i+1}^{1}) \frac{G_p}{2} - \bar{y}_i^1}{-\frac{l_{r1}}{2} - \bar{x}_i^1} \right) \right| \tag{11.26}$$

Because ${\mathbf{F}_2}^{(i)}$ that is obtained in Eq. (11.25) acts on the ith element, this force is divided in $\mathbf{F}_{ei}$ that ith element gains and $\mathbf{F}_{ei+1}$ that i+1th element gains. Therefore each force of nodes can be obtained as

$$\begin{bmatrix} \mathbf{F}_{ei} \\ \mathbf{F}_{ei+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}F_{2x}^{(i)} & \frac{1}{2}F_{2y}^{(i)} & \frac{l_e}{12}F_{2x}^{(i)} & \frac{l_e}{12}F_{2y}^{(i)} & \frac{1}{2}F_{2x}^{(i)} & \frac{1}{2}F_{2y}^{(i)} & -\frac{l_e}{12}F_{2x}^{(i)} & -\frac{l_e}{12}F_{2y}^{(i)} \end{bmatrix}^T \tag{11.27}$$

where $\mathbf{F}_{ei}$ is the force of the ith node, $\mathbf{F}_{ei+1}$ is the force of the i+1th node, and $F_{2x}^{(i)}$ and $F_{2y}^{(i)}$ are components of the x, y direction. As the reaction force of $\mathbf{F}_2^{(i)}$ that the flexible body gains from point A' of rigid body 1, rigid body 1 gains $\mathbf{F}_{r2}$ from the flexible body and moment M_{r2}. This force and moment acting on rigid body 1 can be written as

$$\begin{bmatrix} \mathbf{F}_{r2} \\ M_{r2} \end{bmatrix} = \begin{bmatrix} -\mathbf{F}_2^{(i)} \\ \begin{bmatrix} -\frac{l_{r1}}{2} \\ -\frac{G_p}{2} \end{bmatrix} \times \left(-\mathbf{F}_2^{(i)} \right) \end{bmatrix} \tag{11.28}$$

The constraint between the flexible body and the rigid bodies is expressed by applying Eqs. (11.17), (11.27) and (11.28) to Eq. (11.5).

11.3.3 Control of the System

As a preliminary step, the attitude control by the angle between the arm and the part to be held is considered. The purpose of the proposed method is to control the attitude of the human body by the operating direction of the tether tension by the rotation of the rigid body 1 so that the human moves safely by using the TSMD. The concept of control is shown in Fig. 11.8. This figure is explained here. First, tension of the tether occurs and rotation of the system occurs by the relation between the center of mass of the system and the point of action due to tension. Next, the moment in the opposite direction occurs by the controlling arm (rigid body 1). Then, the rotation is controlled by this moment. In this method, attitude is controlled by feeding the angular velocity and the angle of rotation of rigid body 2 back into the angle of rigid body 1. To control the attitude of the part held by human (rigid body 2), proportional control is used:

$$\theta_{arm} = k\dot{\theta}_{main} \tag{11.29}$$

where θ_{arm} is the angle between body 1 and body 2, $\dot{\theta}_{main}$ is the angular velocity of the part to be held and k is the control gain.

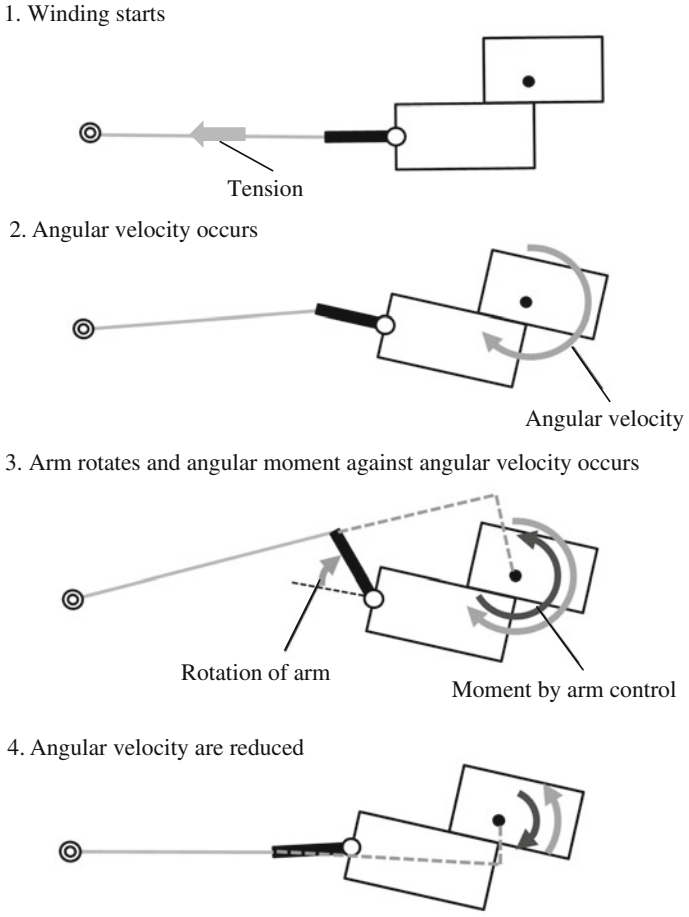


Fig. 11.8 Outline of arm control

11.4 Experiment

11.4.1 Experimental Setup

The experimental equipment is described in this section. The top view of the experimental equipment is shown in Fig. 11.9 and the side view is shown in Fig. 11.10. The experimental equipment has the main unit that is equipped with feature of TSMD, and the unit imitated human body that duplicates the influence of the weight of the human body. The main unit has a motor to wind the tether and a motor for the control arm. The unit imitating the human body has a controller and battery for the motor. An acceleration sensor and a gyro sensor are added in the main unit to measure the motion and to control the posture of main part. The unit imitating the human body has a battery. The mass of the battery shifts the center of gravity of the system.

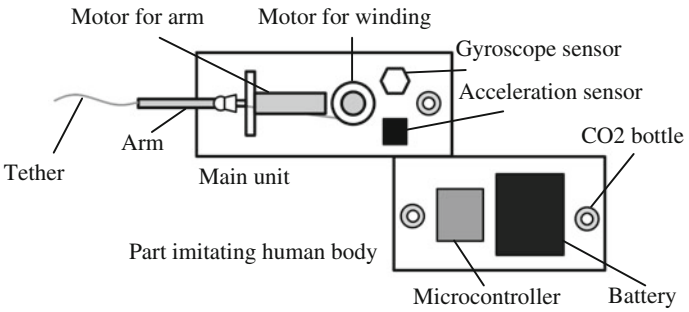


Fig. 11.9 The experimental setup of the TSMD (*top view*)

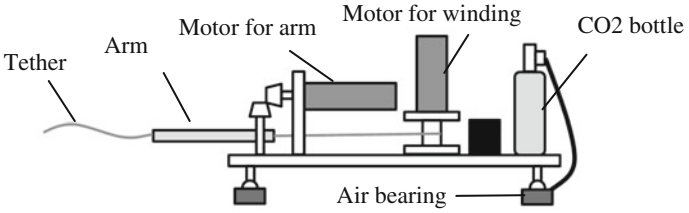


Fig. 11.10 The experimental setup of the TSMD (*side view*)

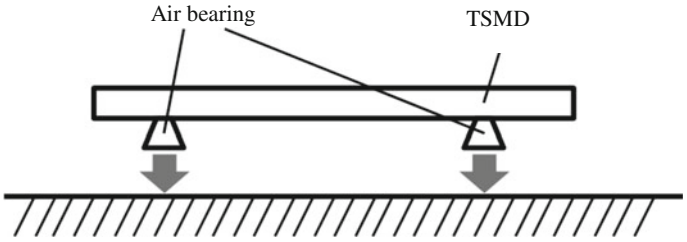


Fig. 11.11 Constitution of two-dimensional microgravity simulator

Therefore, rotation by tension occurs easily. A microcomputer (H8-3052), which is proven as a controller under microgravity, is loaded and gives instructions based on the sensor input to each motor loaded in the main unit. In this chapter, the shooting and attaching actions of the tip of the tether are not considered and are an issue to be addressed in the future.

As a primary consideration of the TSMD, the situation of two-dimensional zero gravity is produced experimentally by an air bearing. The air bearing is used in the experiment using a space robot [17, 18]. In Fig. 11.11, the outline of floating by the air bearing is shown. The air bearing allows air from the CO_2 bomb to enter the contact surface between the TSMD and the flight table. Therefore, this system reduces friction force by the floating TSMD. The flight table is covered by smooth glass with little friction.

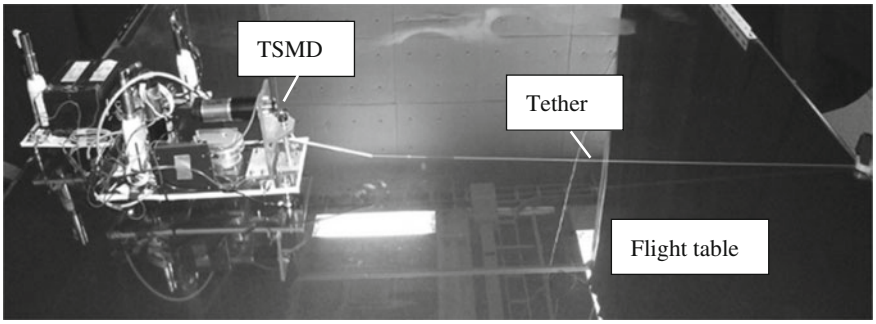


Fig. 11.12 Initial condition of the TSMD

Table 11.1 Parameters of the experimental set up

Length of tether (m)	1.35
Diameter of tether (m)	0.00052
Material of tether	Nylon
Length of the main unit of TSMD (m)	0.43
Length of arm (m)	0.3
width of the main part of TSMD (m)	0.17
Length of part imitating the human body (m)	0.41
Width of part imitating the human body (m)	0.2
Weight of TSMD (kg)	12.1
Winding speed (m/s)	0.143
Arm control gain	0.36

11.4.2 Experimental Condition

In this experiment, the influence of the proposed control method is investigated. Then, the angular velocity of the main unit is examined. The grasping part is attached to the target point. Figure 11.12 shows the initial conditions of the experiment. The initial conditions of the experiment are as follows: the angles of the tether and the arm are 0 [rad], the angle of the main unit is 0 [rad]. The initial conditions are set so that the tether and the arm are straight. The parameters of the TSMD are shown in Table 11.1.

11.5 Discussion of Motion of the TSMD

First, the experimental result and the numerical result of the angular velocity of the TSMD without control are compared. Figure 11.13 shows the time history of the angular velocity of the main unit. In Fig. 11.13, the angular velocity increases in

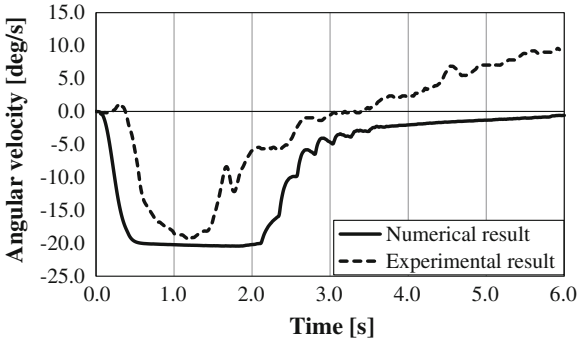


Fig. 11.13 Time history of angular velocity of the TSMD

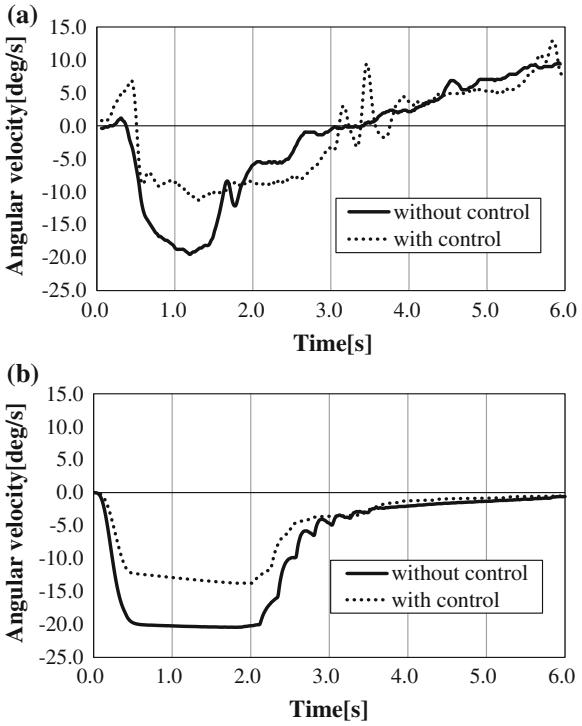


Fig. 11.14 Time history of angular velocity of the TSMD. **a** Experimental result. **b** Numerical result

the counterclockwise direction first. This phenomenon is caused by the tension of the tether. The tension of the tether rotates the main unit. Then, the angular velocity increases in the clockwise direction. This phenomenon is caused by the relationship between the inertia force of the TSMD and the deflection of the tether by winding.

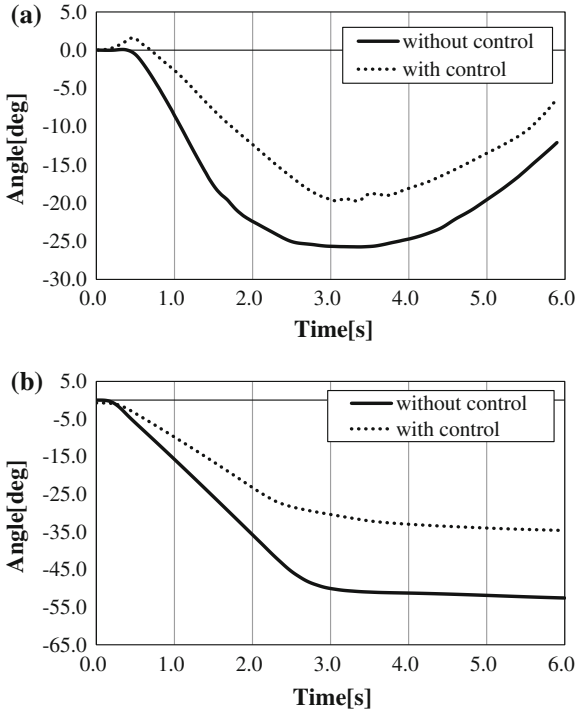


Fig. 11.15 Time history of angle of the TSMD. **a** Experimental result. **b** Numerical result

The angular velocity of both results tends to change in a similar manner. It is assumed that the difference between the numerical result and the experimental result is due to the friction of the flight table and the gravity on the tether, because the TSMD rotates smoothly in the numerical result. Furthermore, the start of motion is delayed in the experimental result.

Next, the influence of the proposed control is investigated. Figure 11.14 shows the time history of angular velocity with control and without control. Figure 11.15 shows the time history of the angle with control and without control. In these figures, control of the arm reduces the peak of the angular velocity. This tendency of the experimental result corresponds to that of the numerical result. Therefore, it is understood that the numerical model is valid for the coupling motion between the tether and the main unit. But, this numerical model has the problem of quantitative correspondence.

Next, the coupling motion of this system is investigated using the result of the numerical simulation. Figure 11.16 shows the shape of TSMD in 0.5-s intervals of the numerical simulation. In the initial stage, rigid bodies are rotated when the tether is winded. At 0.5 s, it is confirmed that the tether is deformed in the case of without control. This phenomenon is caused by the relationship between the initial tension of the tether and the direction of the tether. On the other hand, it is confirmed that the arm rotates and the tether is straight shape under control. From these results, it is supposed that initial rotation is controlled by the arm. Then, it is found that the

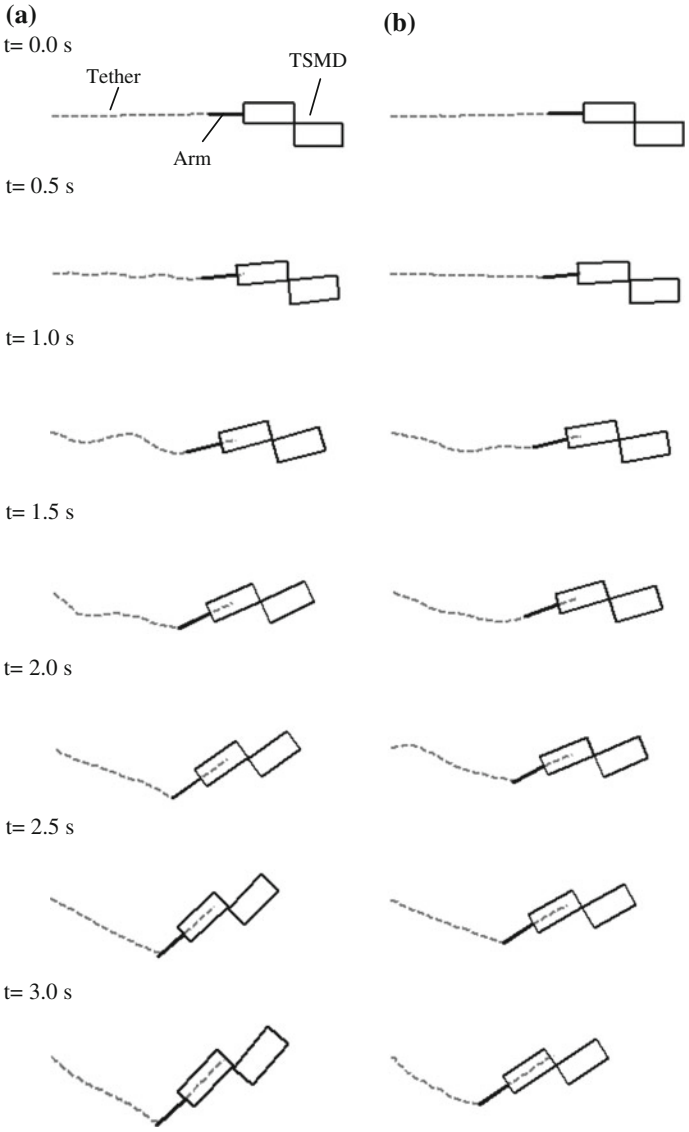


Fig. 11.16 Shape of TSMD at each time **a** without control, **b** with control

tether is deformed. Then, it is found that rotation of rigid body 2 continues until 2 s to that the tether become straight shape. Moreover, it is found that amount of rotation is small in the result with control.

Next, tension of the tether is investigated. Time histories of tension of the tether and contact force at the edge of the arm are shown in Figs. 11.17 and 11.18. First, from the result in Fig. 11.17 it is found that there is difference in amplitude of tension

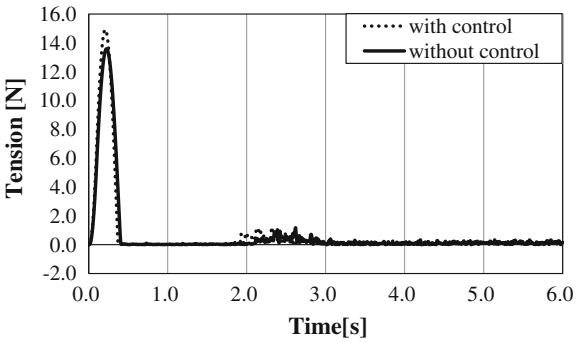


Fig. 11.17 Time history of tension of the tether

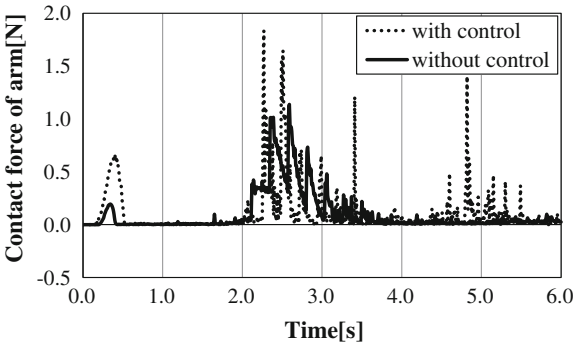


Fig. 11.18 Time history of contact force at the edge of arm

that occurs early. In addition, the variation of tension is confirmed between 2 and 3 s. This is found also from the shape of the tether in Fig. 11.16. It is considered that the difference of action of tension is caused by rotating the arm. In Fig. 11.18, contact force acts on the edge of the arm around 0.3 s and the magnitude of contact force is lager with control. From these results, it is consider that the contact force at the edge is lager by the influence of rotation of the arm. It is found that the tether and the arm contacts by the control of the arm because there are many peaks after 2.0 s with control. From these results, it is found that this system can possibly be a safe mobility device. However, this system needs not only attitude control but also winding control.

11.6 Conclusion

In this research, modeling and formulation of a proposed mobility system using a tether, called the "Tether Space Mobility Device" (TSMD), are conducted. The modeling is formulated by absolute nodal coordinate formulation and the equation of constraint combining flexible body motion and the rigid body motion. For modeling

the winding motion, contact force is used. An experiment of the TSMD under two-dimensional microgravity is conducted.

In the results, it is shown that the angular velocity first increases in the counterclockwise direction. Then, the angular velocity increases in the clockwise direction. The angular velocity of both results tends to change in a similar manner. Furthermore, it is found that the control of the arm reduces the peak of the angular velocity. This tendency of the experimental result corresponds to that of the numerical result. Therefore, it is understood that the numerical model is valid for the coupling motion between the tether and the main unit. However, the numerical model has the problem of quantitative correspondence. Finally, the coupling motion of this system is investigated. It is found that this system can possibly be a safe mobility device, even though it needs not only attitude control but also winding control.

References

1. Modi VJ, Misara AK (1979) On the deployment dynamics of tether connected two-body systems. Acta Astronutica 6:1183–1197
2. Kokubun K, Fujii HA (1996) Deployment/retrieval control of a tethered subsatellite under effect of tether elasticity. AIAA J Guidance Control Dyn 19:84–90
3. Pradhan S, Modi VJ, Misara AK (1999) Tether-platform coupled control. Acta Astronutica 44(5):243–256
4. Nohmi M, Yoshida S (2004) Experimental analysis for attitude control of a tethered space robot under microgravity. Space Technol 24(2–3):119–128
5. Nohmi M, Yamamoto T, Andatsu A, Takagi Y, Nishikawa Y, Kaneko T, Kunitom D (2009) Kagawa satellite "STARS" in Shikoku. Trans Jpn Soc Aeronaut Space Sci 7(odyd26): Tu_7–Tu_12
6. Watanabe T, Fujii HA, Sahara H, Kojima H, Takehara S, Yamagiwa Y, Sasaki S, Abe T, Tanaka K, Oyama K, Johnson L, Khazanov V, Sanmartin RJ, Charro M, Kruijff M, van der Heide JE, Garcia Q, Francisco J, Trivailo P, Williams P (2012) T-Rex: bare electro-dynamic tape-tether technology experiment on sounding rocket S520. J Space Technol Sci 26:14–20
7. Takehara S, Terumichi Y, Nohmi M, Sogabe K (2008) Numerical and experimental approaches on the motion of a tethered system. J Syst Des Dyn 2(5):1106–1117
8. Shabana AA (1996) Finite element incremental approach and exact rigid body inertia. ASME J Mech Des 118:171–178
9. Shabana AA, Hussien HA, Escalona JL (1998) Application of the absolute nodal coordinate formulation to large rotation and large deformation problems. ASME J Mech Des 120:188–195
10. Berzeri M, Campanelli M, Shabana AA (2001) Definition of the elastic force in the finite-element absolute nodal coordinate formulation and the floating frame of reference formulation. Multibody Syst Dyn 5:21–54
11. Sugano N, Imanishi E, Honke K (2001) Dynamic analysis of planar beams moving along the axial direction. Trans Jpn Soc Mech Eng Ser C 67:37–42
12. Kobayashi N, Komaki Y, Watanabe M (2001) Effect of gap size at inlet on spaghetti problem. Trans Jpn Soc Mech Eng Ser C 67:641–647
13. Sugiyama H, Kobayashi N, Komaki Y (2005) Modeling and experimental methods for the dynamic analysis of the spaghetti problem. ASME J Vibr Acoust 127:44–51
14. Kawaguti K, Terumichi Y, Takehara S, Kaczmarczyk S, Sogabe K (2007) The study of the tether motion with time-varying length using the absolute nodal coordinate formulation with multiple nonlinear time scales. J Syst Des Dyn 1(3):491–500

15. Takahashi Y, Shimizu N (2001) Study on the elastic force for the deformed beam by means of the absolute nodal coordinate multibody dynamics formulation (derivation of the elastic force using finite displacement and infinitesimal strain). Trans Jpn Soc Mech Eng Ser C 67:626–632
16. Takahashi Y, Shimizu N, Suzuki K (2003) Study on the damping matrix of the beam element formulated by absolute nodal coordinate approach. Trans Jpn Soc Mech Eng Ser C 69:2225–2232
17. Komatsu T, Uenohara M, Iikura S, Miura H, Shimoyama I (1990) The development of an autonomous space robot operation tested: ASROT. J Robot Soc Jpn 8(6):712–720
18. Nakamura Y, Watanabe Y, Araki K (1999) Experiments of a space robot in free-fall microgravity—experimental methods and visual measurement in a small space. J Robot Soc Jpn 17(7):974–982

Chapter 12
Design Methodology of a Complex CKC Mechanical Joint with an Energetic Representation Tool "Multibond Graph": Application to the Helicopter

Benjamin Boudon, François Malburet and Jean-Claude Carmona

Abstract Due to the operation of the rotor, the helicopter is subject to important vibration levels affecting namely the fatigue of the mechanical parts and the passenger comfort. Suspensions between the main gear box (MGB) and the fuselage help to filter theses problematic vibrations. Their design can be difficult since the filtering should be efficient for different types of external forces (pumping force and roll/pitch torque) which may appear during the flight. As passive solutions classically show their limits, intelligent active solutions are proposed so that the filtering can be adjusted according to the vibration sources. Such studies still suffer from a lack of tools and methods, firstly, necessary to the design of complex mechanical systems (due to their multi-phase multi-physics multi-interaction characteristic, …) and secondly, to develop of an intelligent joint. The main objective of this chapter is to provide a methodology for designing and analyzing an intelligent joint using an energetic representation approach: the multibond graph (MBG). This method is applied here to a complex mechanical system with closed kinematic chains (CKC) which is the joint between the main gear box (MGB) and the aircraft structure of a helicopter. Firstly, the MBG method is analyzed. Secondly, after a brief state of art of the MGB-Fuselage joint, developments focus on the 2D and 3D modeling of the MGB-Fuselage joint with a MBG approach. The 20-sim software is used to conduct the simulation of bond graph. Finally, the MBG models results are presented, illustrating the potential of the MBG tool to predict the dynamic of a complex CKC mechanical system.

B. Boudon (✉) · F. Malburet · J.-C. Carmona
CNRS, LSIS, Arts-et-Métiers ParisTech, 2 Cours des Arts-et-Métiers,
13617 Aix-en-Provence, France
e-mail: benjamin.boudon@free.fr

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_12,

12.1 Introduction

The rotor of a helicopter is a powerful vibration generator that can generate various vibration phenomena. Let us consider:

- forced vibrations,
- resonances "ground and air",
- dynamic problems of the power chain.

Blades undergo periodic and alternating aerodynamic forces whose fundamental frequency is the rotation frequency of the rotor. This result is explained in [1]. These efforts on the blades cause forces and moments on the hub which then becomes a mechanical excitation of the fuselage. Therefore, its behavior depends on its dynamic characteristics and the filtering systems placed between the rotor and the fuselage (as shown in Fig. 12.1). In this sequel, we will focus on one of these filtering systems: the Dynamic Anti-Resonant Vibration Absorber system (DAVI) (called Suspension Antivibratoire à RésonateurIntégré (SARIB) in French).

Figure 12.2 summarizes the main consequences of forced vibrations in a helicopter as explained above. For simplicity, the various couplings between the rotor and the fuselage, due to the actions of the fuselage on the dynamics of the blades, are not taken into account.

The MGB-Fuselage joint must ensure several important functions. Firstly, the joint allows the transmission of the static force necessary to the sustentation of the helicopter with a limited required static displacement. Moreover, the joint helps to reduce the mechanical vibrations transmitted to the fuselage according to the force and displacement aspects. Classically, the MGB-Fuselage joint is composed of four MGB bars and a main membrane as shown in Fig. 12.3.

Different technical solutions exist for the realization of this joint. In this chapter, the SARIB system is particularly studied. The architecture of this system is detailed in Sect. 12.3.

The design and the analyze of such complex mechanical systems are usually conducted with analytical methods based on physical equations or signal-flow method based on transfer functions written on a block diagrams form. Unfortunately, these two classical approaches may cause a loss of the physical sense and the visibility of the modeling assumptions [2]. Moreover, taking account of increasing complexity requires partially to resume a part of the modeling phase.

The "complex mechanical system dynamics" project, funded by European Aeronautic Defense and Space foundation (EADS), focuses on helicopter dynamics and has as main objective the development of an analysis methodology together with the related tools in order to support design and control of such systems. The present chapter presents an energetic representation tool for modeling: multibond graph (MBG). The MBG is applied to model the dynamic of a classical helicopter subsystem: the main gear box MGB-Fuselage joint. This approach enables to represent mechatronics systems in a graphical form describing the exchange of power between basic elements like inertia, compliance, dissipation, conservative power transformation, gyrator actions and sources. The bond graph approach used for multibody system

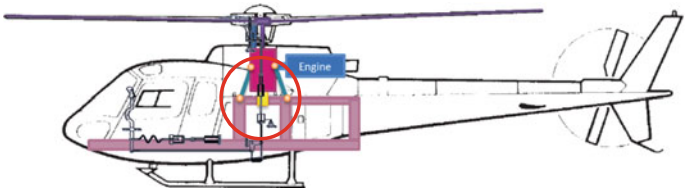


Fig. 12.1 Helicopter suspension between the MGB and the aircraft structure

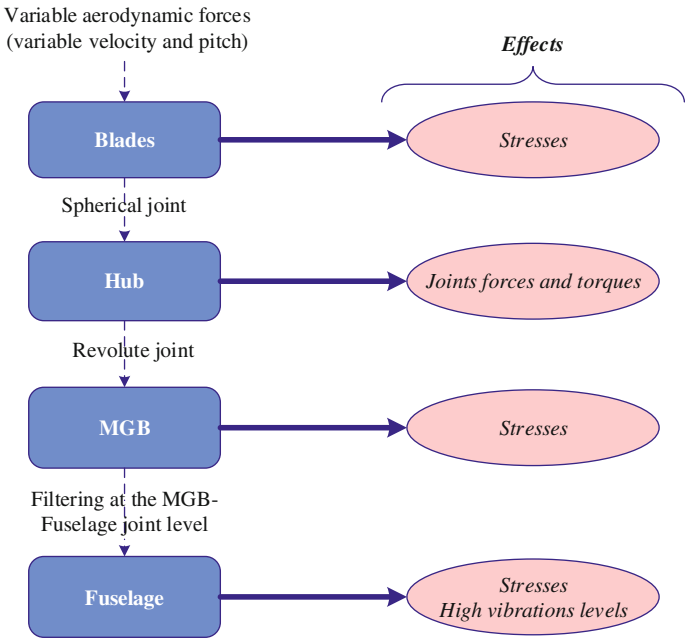


Fig. 12.2 Consequences of the forced vibrations on the helicopter

called multibond graph (MBG) has been introduced by Tiernego and Bos [3], Bos [4]. Library models for a rigid body and for various types of joints have been provided and bond graph models of rigid multibody systems can be assembled in a systematic manner. Further, Marquis-Favre and Scavarda [5] proposed a method dedicated to systematic generation of bond graph models for multibody systems with kinematic loops. Nevertheless, few complex multibody systems with kinematic closed loops have been simulated on dedicated softwares such as 20-sim software [6] (simulation package for dynamic systems using physical components, block diagrams, bond graphs and equations of motion).

The SARIB system here studied is a complex mechanical closed kinematic chain (CKC) system. The dynamics equations of such a CKC system are a differential-algebraic equations system (DAE) which are difficult to treat and which require specific solving methods. It will be shown that the multibond graph method together

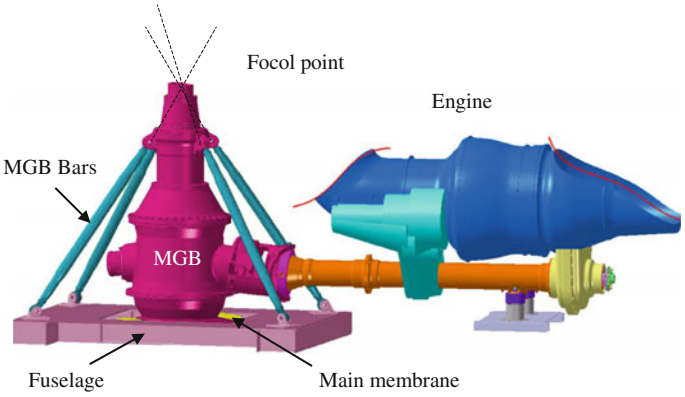


Fig. 12.3 MGB-Fuselage description

with the method of singular perturbations appears to be an elegant and easy solution to derive the simulation of CKC system.

The main objective of this chapter is to present a design methodology based on an energetic representation tool: multibond graph and to show its benefits as a systemic approach. This method will be applied to model a joint between the main gear box (MGB) and the aircraft fuselage which is a complex multi-body system because of the numerous bodies and joints and the mechanical forces applied on the MGB. The chapter is organized as follows. In Sect. 12.2, we shall explain the main advantages of the energetic representation tool chosen: the multibond graph compared to other more classical methods. Section 12.3 describes the kinematic structure and the operation of the MGB-Fuselage joint. The construction of the multibond graph of the joint desired is then detailed in Sect. 12.4. Simulation results will be presented in Sect. 12.5. Finally, conclusions and perspectives will be given in the last Section.

12.2 Energetic Tool: MBG for Complex Mechatronic System Modeling

12.2.1 Characterization of Complex Mechatronic System

The design of the MGB-Fuselage joint studied is within the scope of the design of mechatronic systems. Many definitions of mechatronic systems exist. For example, French standard NF E 01-010 [7] gives the following definition: “approach aiming at the synergistic integration of mechanics, electronics, control theory, and computer science within product design and manufacturing, in order to improve and/or optimize its functionality”.

We shall not detail more the concept of mechatronic systems thereafter. This standard and the chapter [8] have dealt with the issue sufficiently so as to define the perimeter of the mechatronics system studied. However, what we consider as a complex system is going to be defined in a more detailed way in the sequel. The goal is to facilitating a better understanding of the energetic representation tool chosen to describe complex multiphysic systems.

Let us remember that a multiphysic system is a multitechnology system which involves a multidisciplinary approach: mechanics, electronics and control. For the MGB-Fuselage joint equipped with adjustable SARIB system, the presence of control systems and possible electronics devices to achieve energy harvesting widely justify this multiphysic characteristic.

Moreover, a multiphase system is characterized by different operating phases during its life cycle. For example, the joint is built into a helicopter system with many operating phases: on the ground, parking flight, in forward flight.

A multiscale system is characterized by the physical laws of different scales: distributed/lumped parameters and microscopic/macroscopic scale. The MGB-Fuselage joint connection equipped with adjustable SARIB system was modeled as a multibody lumped parameter model in this chapter. Taking into account the nature of certain deformable bodies (as the fuselage) may require additional models with distributed parameters.

A multi interaction system includes a large number of elements in relation to each other and whose interactions can make emerge new properties. This characteristic of complex systems emphasizes their holistic character based on the principle that "the whole is greater than the sum of its parts". The MGB-Fuselage joint equipped with adjustable SARIB system is a system with many bodies constrained by kinematic links. Moreover, this joint is itself embedded in a larger system: the helicopter with which it has many interactions (the fuselage, the rotor, or the command chain …).

Such systems present complex multibehavior (nonlinearity, friction, gap …). In effect, in the case of the MGB-Fuselage joint equipped with adjustable SARIB system, there are primarily geometric nonlinearities and friction in the equations.

12.2.2 Interest of Using a System Approach

Given the multidisciplinary aspects and complexity of mechatronic systems we have stated in the previous paragraph, the design tools must have some essential features to enable their efficient modeling (Table 12.1).

First, the design tool should be based on a unique and unified language for different fields of physics in order to enable a common modeling early in the design phase of multiphysic systems.

Then, the design tool should lead to models describing the physics of the model regardless the purpose of modeling. Such a model having a structure independent of its inputs and its outputs is called acausal. This acausal type of model is particularly interesting to model multiphase systems since the model structure remains

Table 12.1 Complexity of studied system

Design tool characteristics	Complex system characteristics
Unique and unified representation	Multiphysic
Acausal model	Multiphase
Multilevel representation	Multiscale
Modular	Multi-interaction
	Multibehavior

independent of the type of inputs (related to the operating conditions considered) applied to the system.

Then, the design tool should allow a multilevel approach like an object-oriented language. This object-oriented approach facilitates the decomposition of a system into subsystems with the encapsulation of these approaches property. This multilevel approach then permits a better management of two characteristics of complex systems studied. Firstly, it allows a simplification of the presentation of systems with multiple interactions almost essential for their analyses. Indeed, the decomposition of the system into subsystems helps to hide the internal interactions of each subsystem and, therefore, to distinguish the interactions between major subsystems and the internal interactions of these subsystems. Secondly, it facilitates the inclusion of the multiscale aspect since it allows one to encapsulate a distributed parameter model in a lumped parameters model with higher level parameters. However, this last point discussed in [9] remains still under study and has not been validated in this chapter.

Finally, the design tool should provide a modular aspect to the system model as presented in [10]. Indeed, the model must evolve to meet the levels of complexity required for each design problem by the addition or modification of new components and subsystems and by replacing behavior laws. This is intended to deal with all aspects multi-interaction and multibehavior of a mechatronic system.

12.2.3 MBG Modeling

12.2.3.1 Overview

The concept of energy is fundamental in the description of the evolution of technological systems. Energy is present in all areas of physics and is the link between them. From this observation, a number of tools with energetic representation for modeling complex systems have been defined. One of the main tools is the bond graph (BG).

The bond graph was created by Paynter [11] in 1959 and developed by Karnopp et al. [12] at MIT Boston in the United States.

The bond graph is based on a study of the transfer of power in a system modeled by lumped parameters. The bond graph is a graphical modeling tool that covers all

physical systems (mechanical, hydraulic, electronic, thermal …) regardless of their condition (linear, nonlinear, continuous …).

It is represented as an oriented graph showing dynamic variables and power bonds between these variables. The bond graph systematically associate two different variables for each bond: a generalized effort variable (which is a force or a torque in mechanics) and generalized flow variable (which is a translational or rotational velocity in mechanics) on each side of the half-arrow link. Each bond has therefore power information, obtained by the product of these two variables, and allows direct access to the energy transferred by simple integration of power. The bond graph is based on three fundamental types of elements: active element, passive element and junction element. The active elements noted **Se** and **Sf** respectively represent sources of effort and flow. These are the power inputs of the system. The fundamental property that defines a source is that the variable effort (**Se**) or flow (**Sf**) provided by a source to a model is assumed to be independent of the complementary variable flow (**Sf**) or effort (**Se**) which depends on the characteristics of the system and the variable applied. Passive elements **I**, **C** and **R** are the three main components of a bond graph. The first two represent energy storage elements, respectively in kinetic and potential form, while the latter represents a dissipative element. In a bond graph representation of a complete system, these previous elements can be interconnected by connecting elements in common effort (**0** junction) or common flow (junction **1**), or processors elements (**TF**) or gyrators (**GY**) (Table 12.1).

More details on bond graph can be found [13] detailing its construction and operation that we can do.

The bond graph has been extended in the 90s to the study of the multibody systems with three dimensions thanks to the multibond graph formalism. Here, the scalar power bonds become vectors bonds and the elements multiports.

12.2.3.2 Brief Review of MBG

A brief review of multibond graph used for multibody systems is now presented. Readers wishing more details can refer to the multibond graph state of the art directed by Borutzky [14] which is quite exhaustive.

The first works were developed by Tiernego and Bos [3] to model robots. Then Zeid and Chung [15] developed libraries of multibond graph model of three-dimensional kinematic joints. Then Marquis-Favre's [16] contains a large contributions of multibond graph applied to multibody system. The multibond graph is used to model both systems: serial systems and systems with kinematic loops. The concept of word bond graph (WBG) well illustrated in [17, 18] enables to have a more concise and simplified representation. The contributions of Rideout [19] and Rayman et al. [20] present the simulation of multibody system with kinematic loops with multibond graph and 20-sim software. In their work, the method of singular perturbations from the work of Zeid and Chung [15] is applied to allow the simulation of multibody system.

12.2.3.3 Benefits of MBG as a Structural Representation

In the modeling phase, over the last 20 years, new tools based on structural approach have emerged in comparison with the classic functional approach. We can mention monophysic tools such as SPICE for electrical field, ADAMS or LMS for multibodies field and multiphysics tools such as MapleSim software, Modelica or multibond graph. The definition of the scope of structural and functional models can be found in [21]. The MBG approach belongs to this category of the tools enabling the construction of structural models as mentioned in Fig. 12.4. Consequently, it benefits from the same advantages.

This new structural approach as the multibond graph facilitates the systemic approach necessary to design a mechatronics system.

Firstly, the complexity of the system is taken into account more progressively than with conventional analytical techniques since the possible modular approach [10] makes it easier modeling a complex system subsystem by subsystem. Indeed, the modularity allowed by the MBG method enables to make the model evolve to meet the levels of complexity required for each design problem by the addition or modification of new components and subsystems and by replacing behavior laws. As a consequence, the global representation of the system built from subsystems facilitates the management of interactions and/or couplings.

Secondly, the multilevel representation of the system realized thanks to the use of word bond graph (WBG) allows to concatenate the bond graphs of bodies and joints. This technique makes possible to "zoom in/out" on different parts of the system as it can be done in a Simulink model [22].

Consequently, the modular and multilevel aspects of this tool, essential for a systemic approach, help to simplify the representation and analysis of complex systems.

Moreover, the structural approach generally enables the generation of acausal model which makes its structure independent of its inputs and outputs as we mention in Sect. 12.2.2.

12.2.3.4 Methods Comparison: MBG Versus Others Structural Approach

In comparison with others structural tools, new interesting features naturally appears as mentioned in Fig. 12.4.

Firstly, this tool allows engineers and researchers working in multidisciplinary fields (especially mechanics and electronics) to have a unified representation showing power transfer between system's elements in order to support complex multiphysic system modeling. Indeed, it should enable to easily introduce an electronic model of energy harvester or active control system to the mechanical system thanks to the same modeling representation.

Secondly, the classical functional approach using signal-flow can complete the structural multibond graph model in the 20-sim software. This hybrid feature is very useful for performances evaluation and for the determination of a possible control law.

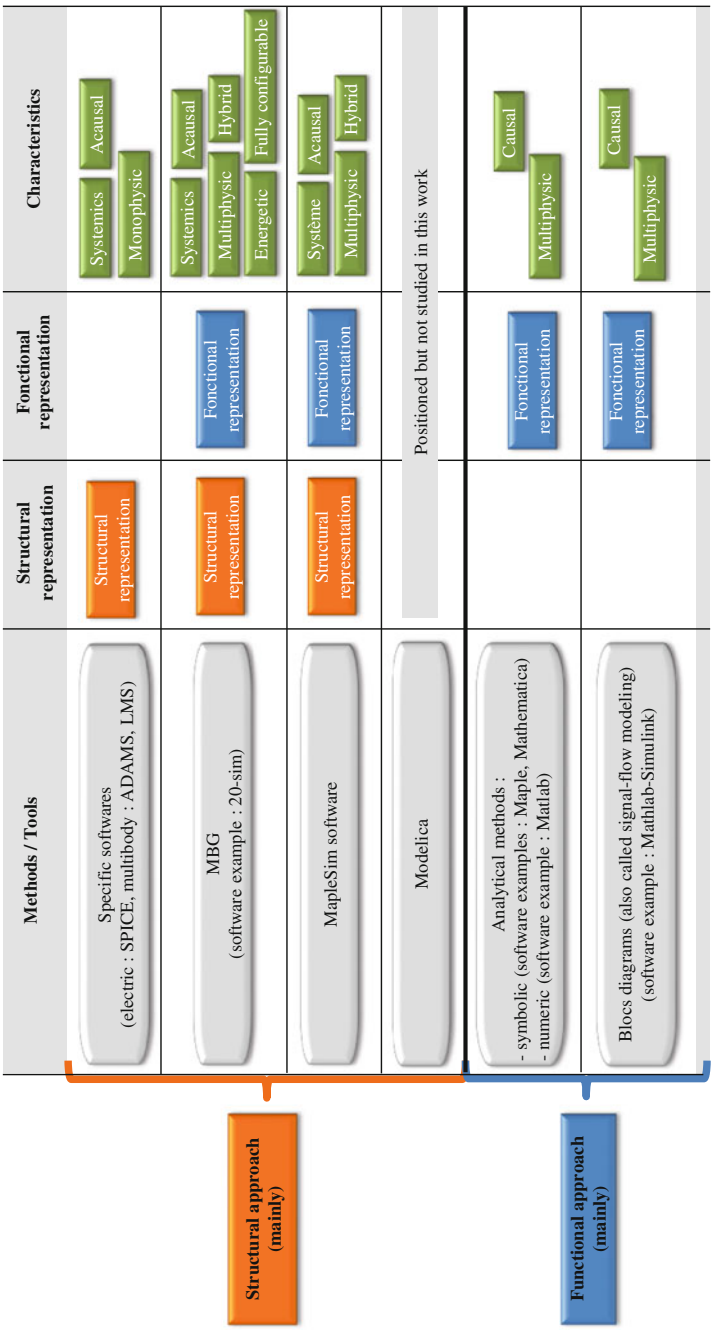


Fig. 12.4 Position of multibond graph method among some design tool

Thirdly, this approach allows to describe the exchange of power or energy between the different components of a mechatronics system. Thanks to multibond graph representation, this energetic approach should permit to analyze the location where mechanical energy is optimum for energy harvesting consideration for example.

Fourthly, contrary to dedicated software enabling a structural approach in multibody modeling where the multibody elements have finite possibilities of parametering, the MBG is more completely configurable since the designer uses the multibond graph of bodies and joint built from the standard elements depicting physic laws and which can be thus easily modified.

12.3 Study Case: The MGB-Fuselage Joint of an Helicopter

The classical MGB-Fuselage joint is composed of four MGB bars and a main membrane as we can see in Fig. 12.5. Let us analyze the principle of operation of this joint. The components of the mechanical actions of the {Rotor + MGB} on the fuselage are composed by a static part of the effort required to the lift and a dynamic part from dynamic excitations induced by the rotor on the fuselage due to its own rotation.

The MGB bars can suspend without flexibility the fuselage to the rotor and thus transmit the lift from the rotor to the structure. In addition, the MGB bars allow the MGB to have a rotation around a point called the focal point which is the point of intersection of the MGB bars.

The membrane is a flexible suspension with some particularities:

- a low stiffness for angular movements on the roll and pitch axes and the linear vertical pumping displacement,
- a very high stiffness for linear movements perpendicular to the vertical direction and for the yaw movement.

Thus, the membrane allows the angular movement of the MGB around the pitch and roll axes. The flexibility of the membrane around these axes allows a strong filtration of the dynamic moments around these axes. This filtering is achieved by adjusting the frequency of the pendulum system smaller than the excitation frequency of the rotor.

In addition, the membrane transmits to the main rotor torque thanks to its very high stiffness around the yaw axis.

In conclusion, the conventional suspensions allow filtering pitch and roll dynamic moments without filtering the pumping dynamic efforts.

The purpose of the SARIB suspension is to render possible the filtering of these pumping dynamic efforts.

The SARIB system is composed of SARIB Bars with a tuning mass on each bar which are installed between the MGB bars and the fuselage. The SARIB system is designed so as to create inertial forces on the fuselage opposite to the force of the

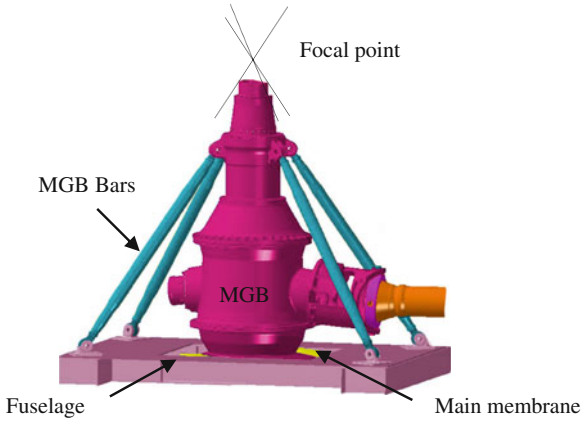


Fig. 12.5 Flexible classical *MGB-Fuselage* joint

membrane. This system enables to reduce the efforts transmitted to the fuselage for a frequency called anti-resonance frequency.

To begin with, the analysis based on multibond graph focuses on the kinematic scheme of the 2D MGB-Fuselage joint. This simplified model is sufficient to identify physical anti-resonance phenomenon.

The kinematic scheme of the 2D MGB-Fuselage joint is composed of four bodies (the MGB, a MGB bar, a SARIB bar and the fuselage considered as fixed) and five joints (three revolute joints and two prismatic joints) as shown in Fig. 12.6. These bodies are assumed to be rigid. Some local moving reference frames are attached to these bodies:

$R_{MGB} = (O_{MGB}, \vec{x}_{MGB}, \vec{y}_{MGB}, \vec{z}_{MGB})$ attached to the MGB,
$R_{BS} = (O_{BS}, \vec{x}_{BS}, \vec{y}_{BS}, \vec{z}_{BS})$ attached to the SARIB Bar,
$R_{BB} = (O_{BB}, \vec{x}_{BB}, \vec{y}_{BB}, \vec{z}_{BB})$ attached to the MGB Bar,
$R_F = (O_F, \vec{x}_F, \vec{y}_F, \vec{z}_F)$ attached to the fuselage.

The orientation of the SARIB Bar and the MGB bar are described respectively by angles γ and ψ. The flexible membrane located between the MGB and the fuselage is modeled with two prismatic joints in serial. The intermediate part (called Int) is considered with negligible mass. The position of MGB is described by x and z coordinates.

Moreover, three springs enable the system to have a good filtration behavior. A torsional spring leads to the limitation of the high movement of the SARIB bar. A weak spring along z axis permits the vibrations filtering. A high spring along x axis prevents from a hyperstatic system.

Next, the simulation of the 3D MGB-Fuselage joint with the same energetic approach will be done. A kinematic scheme of the complete MGB-Fuselage is shown in Fig. 12.7. The joint consists of a fuselage considered as fixed, a MGB and four identical legs and a membrane. Each leg consists of a SARIB Bar and a MGB Bar connected by a spherical joint. The upper end of these legs are connected to the MGB

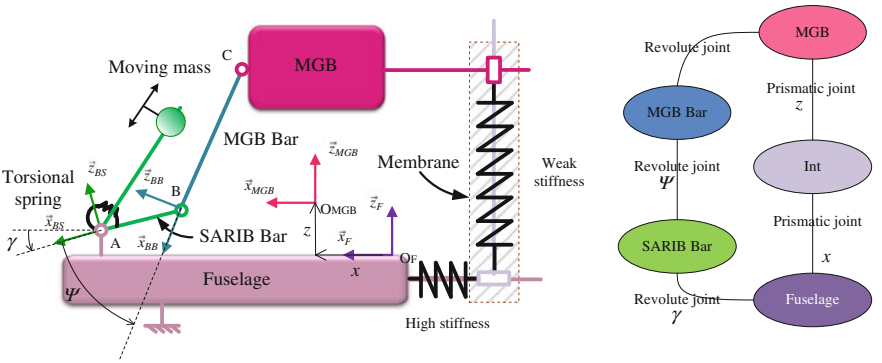


Fig. 12.6 Kinematic scheme of the 2D joint between the main *gear box* and the *fuselage*

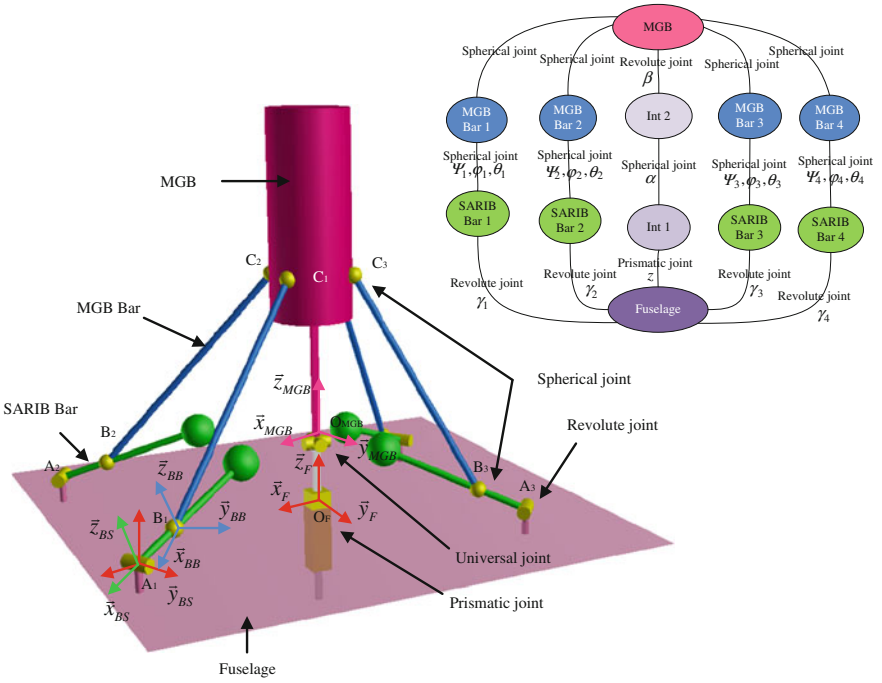


Fig. 12.7 Kinematic scheme of the 3D joint between the main *gear box* and the *fuselage*

with spherical joints and the lower end of these legs are connected to the fuselage through revolute joints. The flexible membrane located between the MGB and the fuselage is now modeled with a prismatic joint and two revolute joints in serial. The intermediate parts (called Int1 and Int2) are considered with negligible masses. For simplicity, the orientation and position parameters are not represented in the kinematic scheme but are described in the joints graph.

12.4 MBG Modeling

12.4.1 Modeling a Multibody System

Only the theoretical elements essential to the practical realization of the multibody bond graph representation is developed.

Bond graph construction based on multi-body dynamics equations can be established either with the fundamental principle of dynamics (or Newton–Euler equations) or by using Lagrange equations. Depending on the starting point, several bond graph construction methods have been developed:

- the "Tiernego and Bos" method from the application of the fundamental principle of dynamics,
- the "Karnopp and Rosenberg" method from the application of the Lagrange equations.

The method used in this chapter is the method of "Tiernego and Bos" since it allows to describe the system as an assembly of subsystems composed of bodies and joints. This assembly of sub-systems is clearly facilitated by the use of word bond graph (WBG). In the WBG, the bond graph of solids and joints are encapsulated in order to focus only on the relationship between solids and joints. Each word bond graph element (bodies or joints) is linked to another word bond graph element through two power bonds for the rotational and translational power transmissions. Each power bond carries a 3D generalized flow vector (rotational or translational velocity) and the complementary 3D generalized effort vector (torque or force).

Bond graph construction developed with the "Tiernego and Bos" method requires the knowledge of a number of multibond graph elements, the bond graph modeling of a rigid body and joints.

The multibond graph elements (multibond or vector bond, junctions, multiport energy storage elements, multiport transformers and gyrators) are directly used. Readers can refer to [14] to find the details of the modeling of those elements.

12.4.2 Simulation Difficulties of CKC Systems

The simulation of mechanical system with kinematic loops requires specific methods. This difficulty does not come from the multibond graph tool but from the application of dynamics equations to such systems where some kinematic variables are linked together. Regardless the analytical method employed (fundamental principle of dynamics or Lagrange equations with multipliers), the equations obtained are differential algebraic equations (DAEs) whose numerical resolution requires specific numerical integration methods. These difficulties to solve numerically differential equations are developed, for example, in Marquis-Favre [23]. A recent review of the methods for solving DAEs can be also found in [24]. To sum up, one can find

three groups of methods: the direct resolution of the DAE thanks to specific solvers, the reduction of the DAE in an ODE like the Baumgarte stabilization method or partitioning method and the conversion to an ODE by modifying the model system. The singular perturbation method which is used in the chapter belongs to the last category of these methods that is to say the conversion to an ODE by modifying the model system.

12.4.3 Use of the Singular Perturbation Method

The multibond graph simulation with the method of the singular perturbation is quite easy to implement compared to conventional techniques used during an analytical study. Others techniques based on multibond graph enabling to treat the simulation of mechanical with kinematic loops exist and are described in Marquis-Favre's [16]. However, we decided to use the method of singular perturbation which, to our point of view, keeps a physical insight and is the simplest to apply.

The method of singular perturbation consists in augmenting the multibond graph of the joints with parasitic elements [19, 20]: stiffness and damping elements corresponding to C energy store element and R resistive element. The values of the compliant elements must be chosen carefully. To our knowledge, two methods for selecting these elements exist: the eigenvalues decoupling between the parasitic frequency and the system frequency and the use of activity metric [19]. These parameters can be chosen so as to model the joint compliances which exist in all mechanical joints. Thus, this point gives to this method a physical significance. The stiffnesses introduced should be high enough in order not to change the dynamic of the system but not too high so as to prevent the numerical difficulties of stiff problems (with high-frequency dynamics). This method leads to a necessary compromise between the accuracy of the results and the simulation time. Moreover, the stiffer the system is, the more numerical errors are reduced but the simulation time remains important. However, the increase of the simulation time can be balanced by parallel processing as the mass matrix in a block-diagonal form can enable to decouple the system as it is explained in [25]. As T. Rayman recommends, adding a damping element (R resistive element) in parallel with the stiff spring (C energy store element) enables to dampen the high eigenfrequency associated with the high stiffness. The exact influence of these parameters still remains a research work in which the authors are particularly interested in.

If the kinematic constraints modeled by the multibond graph of the joint are rigidly imposed, derivative causality appears at the multibonds connected to the translational inertia elements. The derivative causality due to constraints requires that the equations derived from the bond graph are differential algebraic equations (DAEs). The resolution of such equations is quite complex from a computational point of view as we explained before. The method of singular perturbation enables to relax the kinematic joint constraints. The dynamic equations are in a ODE form

with no geometric constraints to deal with. Consequently, it leads to a bond graph with integral causality which can be simulated easily.

As Rayman explains in [20], Marquis-favre and Scarvada developed the method of "privileged frame" [5] to facilitate the resolution of multibody system with kinematic loops. However, it is important to notice that even this method helps to minimize the number of coordinate transformations required in a multibody model with kinematic loops, it does not fundamentally permit the simulation of this system.

12.4.4 Construction of the MBG Model of the MGB-Fuselage Joint

In this section, the bond graph modeling of the rigid body is first recalled. The bond graph modeling of joints is described since they are modeled with some particularities compared to the classical way of modeling that we can find enabling to simulating serial mechanical system. Indeed, as already explained, the kinematic joints have compliant elements so as to enable the simulation of this system with kinematic closed loops.

12.4.4.1 Rigid Body Modeling

Let us remember (Fig. 12.8) the architecture of a rigid body multibond graph model based on [5, 14, 16, 18].

This bond graph architecture is based on the Newton–Euler equations with respectively the inertia matrix (modeled with a multiport energy store element $\left[\mathbf{I}_{S_i,G_i}\right]_i$ in the upper part) associated with gyroscopic terms (modeled with a multiport gyrator element also called Eulerian Junction Structure about mass-center of body i expressed in its frame $\left[\mathbf{EJS}_{G_i}\right]_i$ and the mass matrix modeled with a multiport energy store element $[\mathbf{m}_i]_0$ in the lower part). The upper part of the MBG represents the rotational dynamic part expressed in the body frame while the lower part is for the translational dynamic part expressed in a inertial reference frame (or Galilean frame). The two corresponding 1-junctions arrays correspond respectively to the angular velocity vector of body i $\vec{\Omega}\,(i/0)^i$ and the translational velocity vector of the center of mass of body i $\vec{V}\,(G_i/R_0)^0$ expressed in these two coordinate frames.

The central part of the MBG describes the kinematic relations between the velocities of the two points of the body i ($\vec{V}\left(M_j/R_0\right)^i$ and $\vec{V}\,(M_k/R_0)^0$) and the velocity of the center of mass $\vec{V}\,(G_i/R_0)^i$ resulting from the formula of the rigid body. The modulated transformation element (MTF) between $\vec{V}\,(G_i/R_0)^i$ and $\vec{V}\,(G_i/R_0)^0$ represents the coordinate transformation between the body frame and the inertial frame.

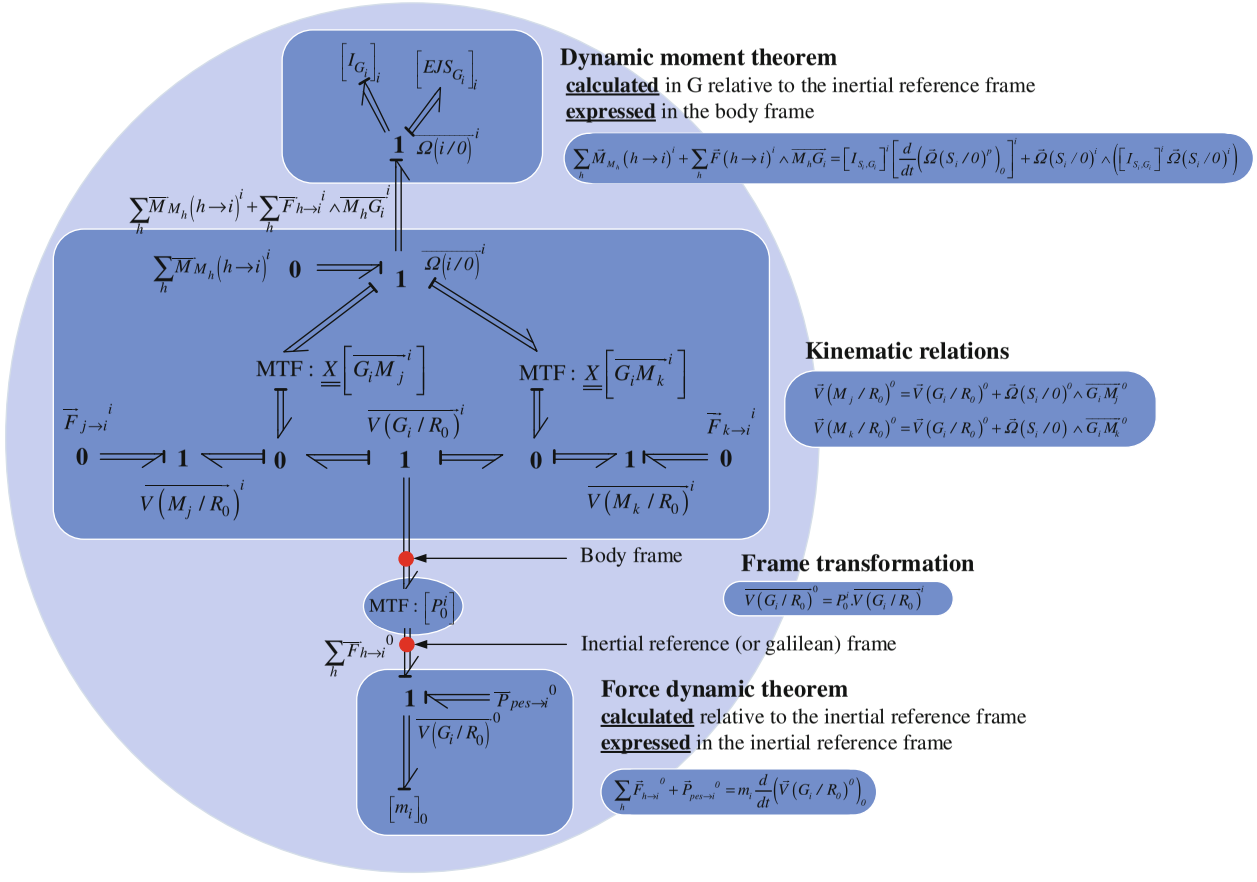


Fig. 12.8 Multibond graph representation of a rigid body i

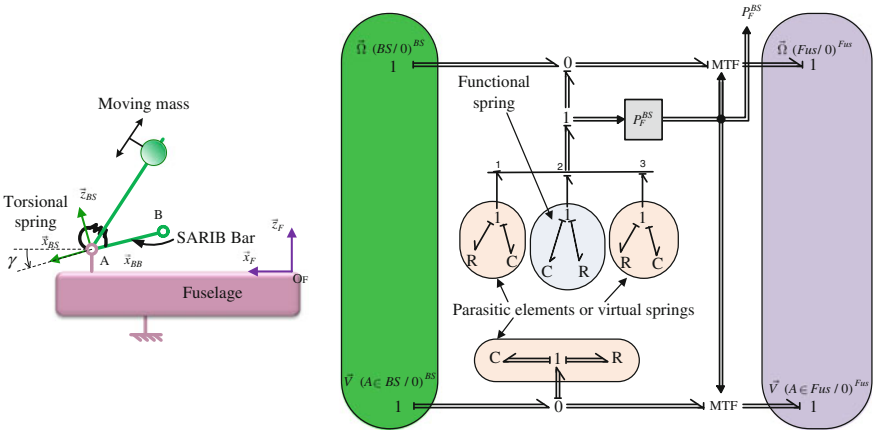


Fig. 12.9 Kinematic scheme and multibond graph of the revolute joint between *SARIB bar* and *fuselage*

12.4.4.2 Kinematic Joints Modeling

For simplicity, only the modeling of the joints necessary for modeling the 2D MGB-Fuselage are here considered: the joint revolute and prismatic including compliant elements are described. On this base, the modeling of different joints such as spherical joint and others needed for the 3D model could be easily derived.

Revolute joint

The kinematic scheme and the multibond graph of the revolute joint between the SARIB Bar and Fuselage are illustrated in Fig. 12.9. In this multibond graph model, the variables used are:

- the angular velocity of the fuselage $\vec{\Omega}\,(Fus/0)^{Fus}$ and the SARIB bar $\vec{\Omega}\,(BS/0)^{BS}$ relative to the inertial reference frame. The subscripts refer to the frames where these velocities are expressed in,
- the translational velocities of the fuselage $\vec{V}\,(A \in Fus/R_0)^{Fus}$ and the SARIB bar $\vec{V}\,(A \in BS/R_0)^{BS}$ relative to the inertial reference frame at the point A,
- the transformation matrix P_F^{BS} determined thanks to the angular velocity as explained in [19].

Prismatic joint

The kinematic scheme and the multibond graph of the prismatic joint along z axis between the MGB and the intermediate body are shown in Fig. 12.10 as follows:

In this multi bond graph model, the variables used are:

- the angular velocity of the MGB $\vec{\Omega}\,(MGB/0)^{MGB}$ and the intermediate $\vec{\Omega}\,(INT/0)^{MGB}$ relative to the inertial reference frame,

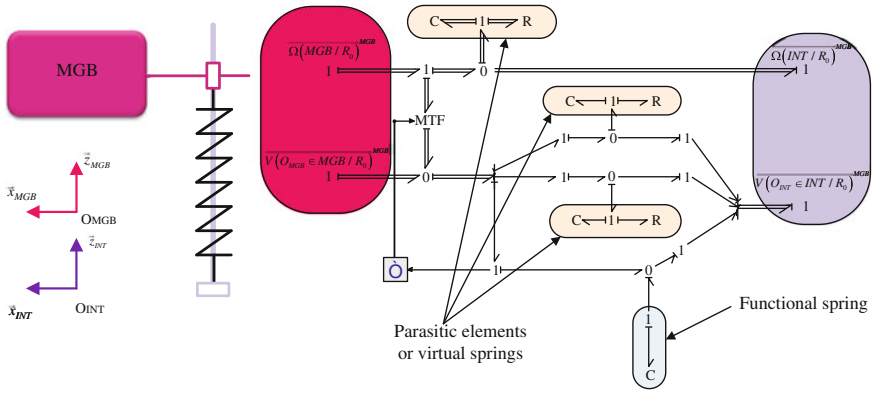


Fig. 12.10 Kinematic scheme and multibond graph of the prismatic joint between *MGB* and intermediate body

- the translational velocities of the MGB $\vec{V}\,(O_{MGB} \in MGB/R_0)^{MGB}$ at the point O_{MGB} and the intermediate body $\vec{V}\,(O_{MGB} \in INT/R_0)^{MGB}$ relative to the inertial reference frame.

12.4.4.3 The Complete Model

Individual models of joints and bodies, previously described, are connected together according to the kinematic scheme as shown in Figs. 12.11, 12.12 and 12.13. The MGB is excited by a vertical periodic force $f\,(t) = F\cos(\omega t)$. In these figures, three types of multibond graph elements can be thus distinguished:

- the rigid bodies, such as the fuselage, the SARIB Bar, the MGB Bar and the MGB.
- the joints, such as revolute joint used between the SARIB Bar and the fuselage.
- the multibond graphs power bonds (half-arrows).

The simulation of the 2D and 3D MGB-Fuselage multibond graphs is then possible.

12.5 Results and Comments

Like for all multibody simulation analyses, the evolution of the different movement parameters of the system may be deduced. For example, the position of MGB gravity center x_{GMGB}, y_{GMGB}, z_{GMGB}, SARIB Bar 1 gravity center x_{GBS_1}, y_{GBS_1}, z_{GBS_1}, the angular parameters of the SARIB Bars $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and the MGB Bars $\psi_1, \psi_2, \psi_3, \psi_4$ are shown in Fig. 12.14. In the same way, forces transmitted to the fuselage joint (revolute joint between SARIB Bar and fuselage) may be deduced immediately.

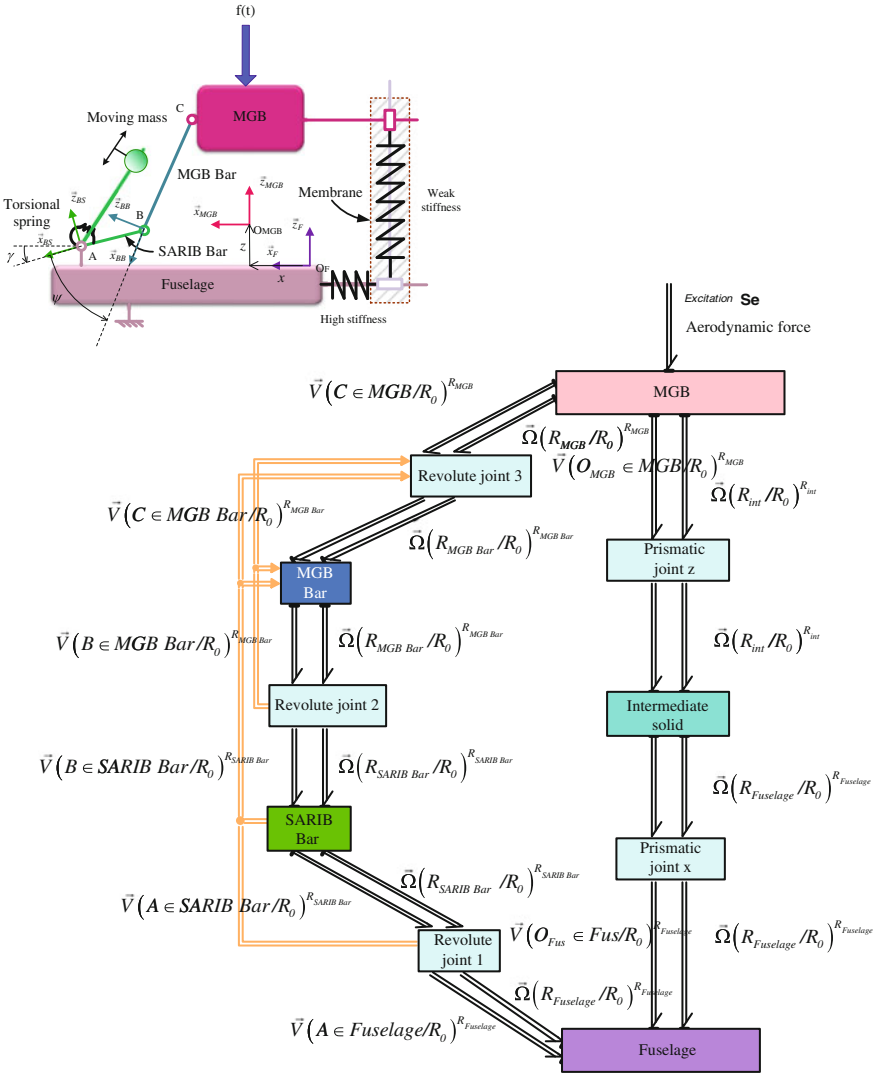


Fig. 12.11 Word bond graph built on 20-sim of the 2D *MGB-Fuselage*

For example, the components f_{A_1x}, f_{A_1y}, f_{A_1z} of the forces applied by the SARIB bars to the fuselage at the A1 point expressed in the fuselage frame are shown in Fig. 12.15. Let us note that, using mechanical analytic methods, some calculations are needed so as to expressjoint forces.

Thanks to complementary tool proposed in 20-sim software, frequency response can be determined after having chosen inputs and outputs.

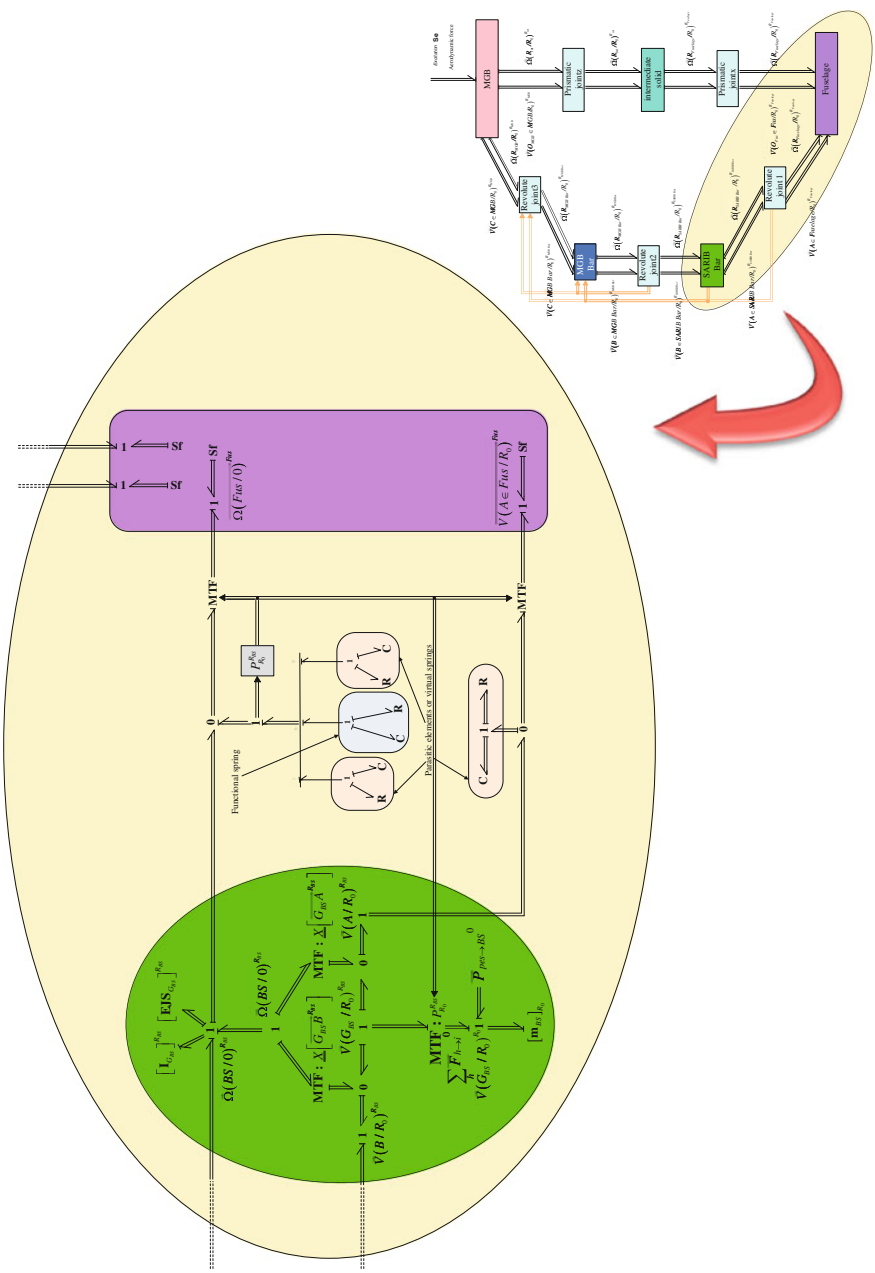


Fig. 12.12 Partially detailed multibond graph of the 2D *MGB-Fuselage* joint

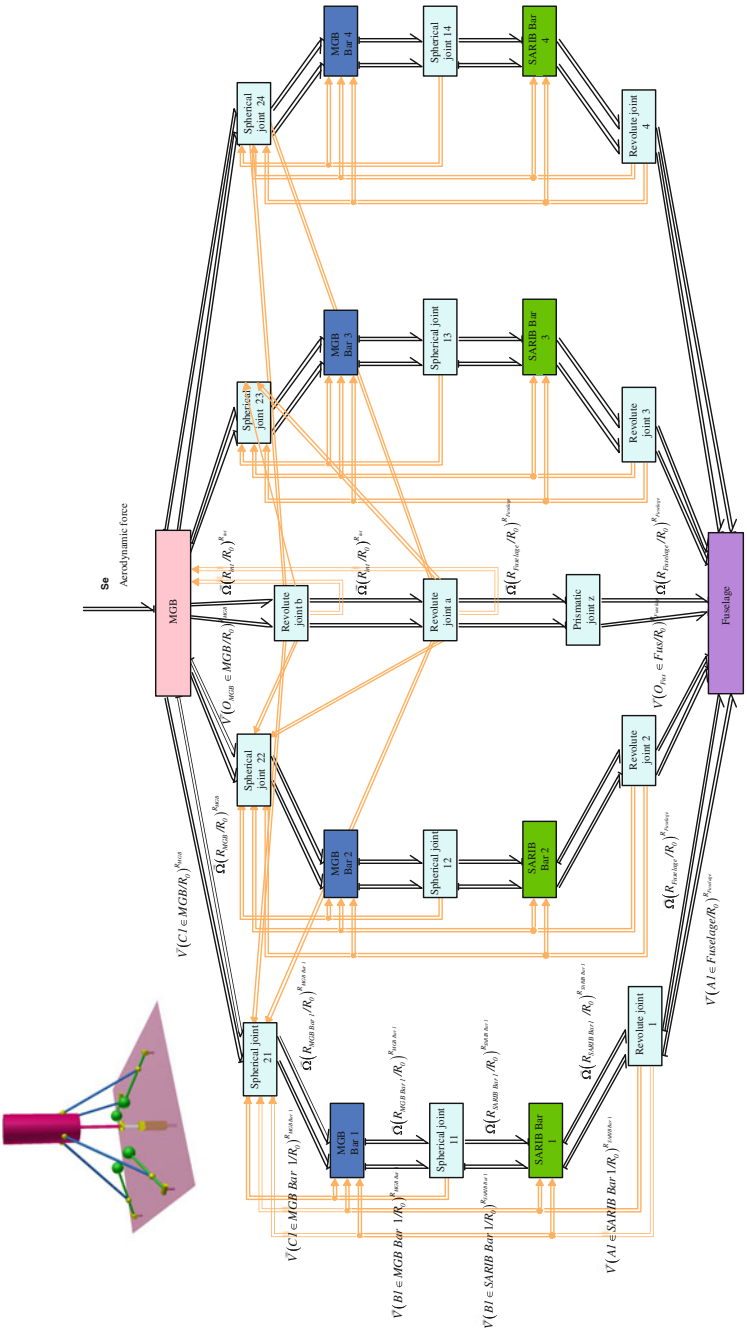


Fig. 12.13 Word bond graph of the 3D *MGB-Fuselage* joint

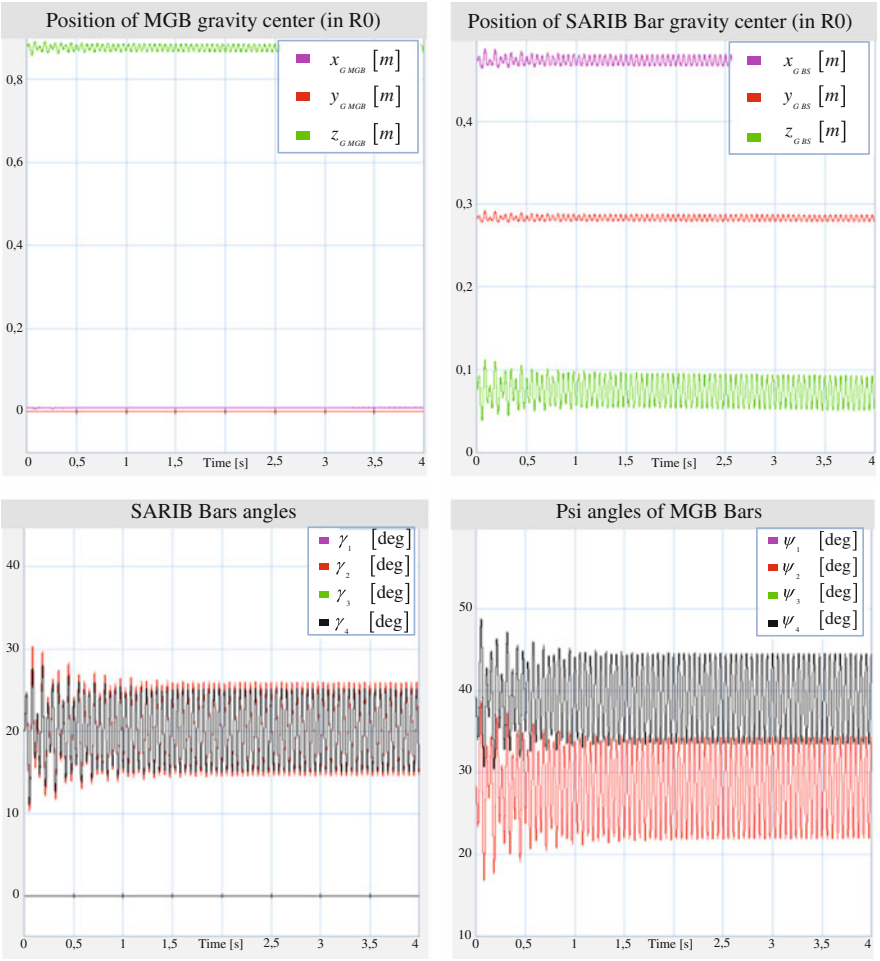


Fig. 12.14 Movements parameters of the 3D model

First, the transmissibility function between the forces transmitted to the fuselage and the excitation force has been deduced for the 2D model. As we can see in Fig. 12.16, the transmissibility presents an anti-resonance frequency. The SARIB Bar plays his role since the joint enables to isolate the fuselage from the force coming from MGB at this specific frequency.

Then, the transmissibilities between the joint reaction at the different revolute joints and the excitation force have been also determined for the 3D model.

Observing the transmissibility curves involving the vertical components of joint reactions (Fig. 12.17 up), we also find the anti-resonance phenomenon that has already been described with the 2D model with antiresonance around 18 Hz.

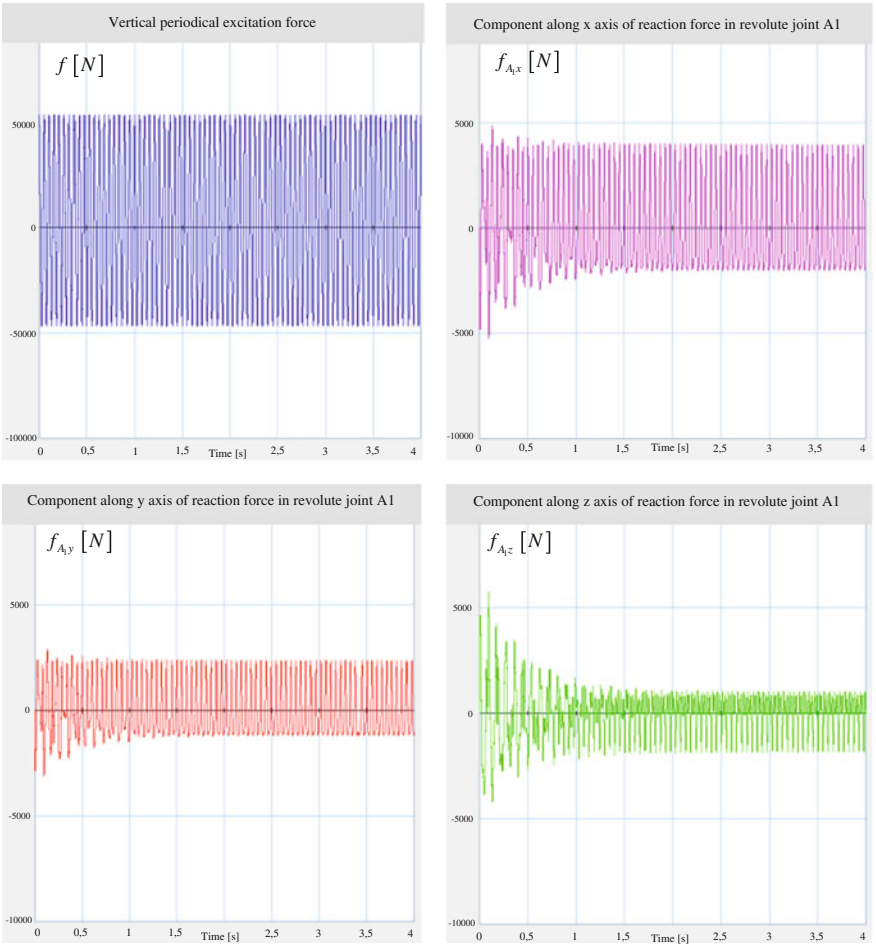


Fig. 12.15 Excitation force and reaction forces in the revolute joints of the 3D model

The layout of transmissibilities, involving components in the xy plane of joint reaction (Fig. 12.17 bottom) shows that the anti-resonance phenomenon does not occur at the same frequency as before. This anti-resonance phenomenon does not occur at the same frequencies according to the type of stress applied to the MGB. This variation of the antiresonance frequency explains the interest of designing an intelligent joint SARIB adapted to external forces applied to the MGB.

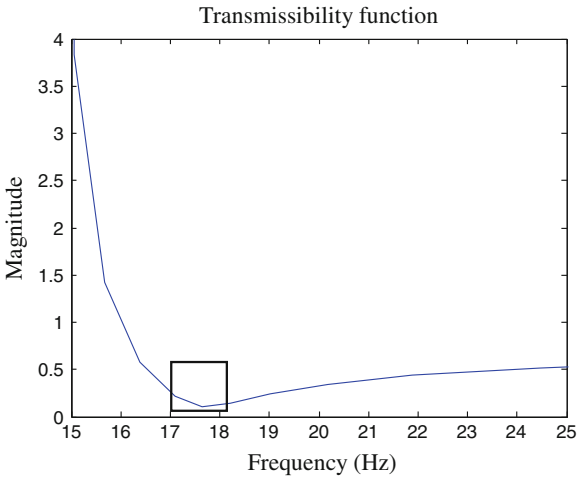


Fig. 12.16 Anti-resonance frequency on transmissibility function

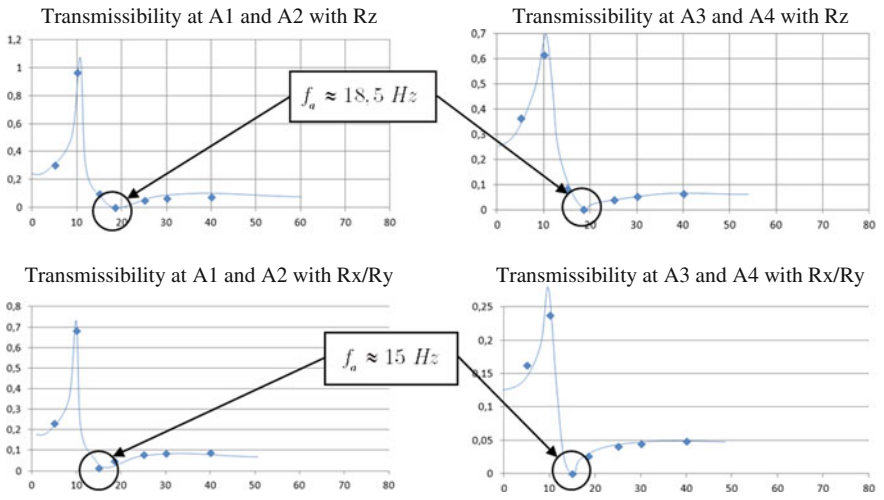


Fig. 12.17 Anti-resonance frequency on transmissibility

This analysis can be also done with an energetic point of view. Indeed, adding power sensors in all connections in the multibond graph, the flow of power can then be evaluated. As expected, all the power provided from the excitation of MGB is sent to the SARIB Bar.

12.6 Conclusion

In this chapter, it is shown how dynamic simulations of an aeronautic complex subsystem can be conducted thanks to a relevant multibond graph representation. In this sense, the proposed contributions consist in providing a relevant methodology to model a multibody system with closed kinematic chains using the bond graph formalism and in comparing this method with others classical methods of modeling.

The proposed methodology is based on three steps. In the first step, the modeler has to build the bond graph of a rigid body. The second step is dedicated to model the different joints connecting the bodies of the system. A fundamental point of this step is the use of parasitic compliant elements for the modeling of the kinematic constraints provided by the joints. The third step treats the assembly of the different created models (rigid bodies and joints). This step can be easily conducted with the help of a well-structured library of components.

The simulation results of the MBG model of the studied joint have been presented. It shows the need to keep a sufficiently complete model so as to predict the anti-resonance phenomenon which exists in this system. Indeed, the 3D model can highlight the existence of different values of anti-resonant frequencies following the direction of efforts observed that had not been visible with the 2D model.

The comparison of multibond graph with others classical methods of modeling shows that this tool appears to be a useful tool for engineers in the context of multibody modeling. The main arguments are now recalled.

Its hierarchical and modular properties enables MBG to be a structural tool. Therefore, the constructed multi-body dynamic models enable to obtain a quite simple representation of a complex system since the multibond graph model highlights the topology of systems. Moreover, the simulation of multibody systems with closed kinematic chains may appear easier to conduct than the classic analytical method. The method of singular perturbation employed in this multi bond graph representation enables to avoid dealing with kinematic constraints equations and consequently to have only ODE systems to solve instead of DAEs. The use of dedicated software such as 20-sim may allow to hide this complex step for the modeler. Finally, we should not forget that the multibond graph is also a unified power based approach which enables to model many multi domains systems and to analyze the description of the energy between the components of such systems.

Future works are being conducted so as to exploit multibond graph models for control design purposes. The investigations will be lead in two directions. The first exploitation of multibond graph representation shall focus on scalar BG analysis. The second exploitation of multibond graph shall lead to control architecture by means of inversion techniques with the help of complementary tool such as energetic macroscopic representation (EMR) designed for this purpose. It should permit to design more robust control laws with less energy consumption.

Acknowledgments This research work received support from the Chair "Dynamics of complex mechanical systems—EADS Corporate Foundation—Arts et Métiers ParisTech and Ecole Centrale de Marseille". Thanks to Paul B.T. Weustink working as Controllab Products for his help on the use of complementary tools of 20-sim software.

References

1. Krysinski T, Malburet F (2006) Mechanical vibrations: active and passive control, Wiley-Iste edn edn. Wiley, New York
2. van Amerongen J, Breedveld P (2003) Modelling of physical systems for the design and control of mechatronc systems. Ann Rev Control 27(1):87–117
3. Tiernego MJL, Bos AM (1985) Modelling the dynamics and kinematics of mechanical systems with multibond graphs. J Franklin Inst 319:37–50
4. Bos AM (1986) Multibody systems in terms of multibond graphs with application to a motorcycle multibody system. University of Twente, Enschede, The Netherlands
5. Marquis-Favre W, Scavarda S (1996) Bond graph representation of multibody systems with kinematic loops. J Franklin Inst 335B(4):643–660
6. http://www.20sim.com/
7. Afnor (2008) Mécatronique—vocabulaire, ed: French Standard NF E01–010. http://www.afnor.org/
8. Isermann R (2008) Mechatronic systems—innovative products with embedded control. Control Eng Pract 16:14–29
9. Lebrun M (2003) Simulation et CAO en automatique et mécatronique. Techniques de l'ingénieur
10. Jardin A (2010) Contribution à une méthodologie de dimensionnement des systèmes mécatroniques: analyse structurelle et couplage à l'optimisation dynamique. Thèse de doctorat, Electronique, Electrotechnique, Automatique EEA, INSA de Lyon, Lyon
11. Paynter HM (1961) Analysis and design of engineering systems. MIT Press edn, Cambridge
12. Karnopp DC, Margolis DL, Rosemberg RC (2000) System dynamics—modeling and simulation of mechatronics systems. Wiley edn, New York
13. Dauphin-Tanguy G (1999) Les bonds graphs et leur application en mécatronique. Techniques de l'ingénieur
14. Borutsky W (2010) Bond graph methodology—development and analysis of multidisciplinary dynamic system models. Springer edn, New York
15. Zeid A, Chung C-H (1992) Bond graph modeling of multibody systems: a library of three-dimensional joints. J Franklin Inst 329:605–636
16. Marquis-Favre W (1997) Contribution à la représentation bond graph des systèmes mécaniques multicorps. Thèse de doctorat
17. Verge M, Jaume D (2004) Modélisation structurée des systèmes avec les bonds graphs. Technip edn, Paris
18. Vaz A, Hirai S (2004) Modeling a hand prosthesis with word bond graph objects. In: Proceedings of international conference on integrated modeling and analysis in applied control and automation (IMAACA 2004)
19. Rideout G (2004) System partitioning and physical domain proper modeling through assessment of power-conserving model structure. Thèse de doctorat, University of Michigan
20. Rayman T, Rideout G, Krouglicof N (2012) Evaluation of dynamic performance of non-spherical parallel orientation manipulators through bond graph multi-body simulation. Presented at the bond graph modeling Genoa, Italy, 2012
21. Chan CC, Bouscayrol A, Chen K (2010) Electric, hybrid, and fuel-cell vehicles: architectures and modeling. IEEE Trans Veh Technol 59(2):589–598
22. http://www.mathworks.fr/products/simulink/
23. Marquis-Favre W (2007) Simulation des mécanismes: résolution des équations dans les logiciels. Techniques de l'ingénieur Modélisation mécanique
24. Uchida TK (2011) Real-time dynamic simulation of constrained multibody systems using symbolic computation. Systems design engineering, University of Waterloo, Ontario
25. Wang J, Gosselin C, Cheng L (2001) Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach. Multibody Syst Dyn 7(2):145–170

Chapter 13
Comparison and Field Test Validation of Various Multibody Codes for Wind Turbine Modelling

János Zierath, Roman Rachholz and Christoph Woernle

Abstract The decrease of fossil energy sources leads to an increased use of renewable energy sources like wind energy. The design of the mechanical components of a wind turbine is considerably governed by their fatigue behaviour over the product life cycle. Therefore, reliable estimations of the interface loads on the components, by means of appropriate multibody models, are necessary. While simplified wind turbine design codes, such as Flex5 or GH Bladed, have been mainly used for previous wind turbine developments, general purpose multibody simulation environments like MSC.Adams, SIMPACK, or alaska/Wind in combination with specific aerodynamic simulation packages are now applied. Here, the components of a wind turbine, such as the drive train with gear pair contacts, a flexible main frame, or a lattice tower, can be modelled in much more detail and specific manner compared to previous simulation models. For example, the geometric nonlinear behaviour of the blades can be taken into account which is essential for the simulation of long slim blade designs. In the present contribution, different multibody packages for wind turbine modelling are compared on the basis of simulation models of an existing wind turbine. Beside of the special wind turbine design code Flex5, developed at the Technical University of Denmark Copenhagen (DTU), the commercial multibody simulation packages MSC.Adams and SIMPACK are used. For the validation of the simulations, extensive measurements on a wind turbine prototype have been evaluated comprising measurement data over a period of more than 1.5 years. To compare

J. Zierath (✉)
W2E Wind to Energy GmbH, Strandstrasse 96, 18055 Rostock, Germany
e-mail: jzierath@wind-to-energy.de

R. Rachholz · C. Woernle
Chair of Technical Dynamics, University of Rostock, Justus-von-Liebig-Weg 6, Rostock, Germany
e-mail: roman.rachholz@uni-rostock.de

C. Woernle
e-mail: woernle@uni-rostock.de

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_13,

measurements and simulations, statistical and dynamical evaluations of the results have been done.

13.1 Introduction

Load calculation is a very important part in the development of wind turbines. Prototype testing of loads acting on a wind turbine in the field is not possible because of its long lifetime. Accurate load calculations are necessary to ensure that the wind turbine withstands the loads during the expected lifetime. An extensive introduction into fundamentals, technologies, application, and economics of wind turbines is provided by [8].

During the past decades Flex5 has been used as a standard tool for load calculation of wind turbines. Ongoing development in the simulation of wind turbines leads to the use of general multibody design codes. This chapter describes the development of a state-of-the-art multibody model of a 2.05 MW wind turbine designed by W2E Wind to Energy using different multibody codes. The prototype of the wind turbine erected in Tarnow, Mecklenburg-Western Pommerania, and the corresponding CAD model are shown in Fig. 13.1.

The general purpose multibody programs MSC.Adams 2012 x64 [12] and SIMPACK 9.3 [17] were chosen to build up the wind turbine model. A detailed description of the drive train model can be found in [14]. The aerodynamic forces are applied to the multibody model using the AeroDyn source code v13.01 which was developed by the National Renewable Energy Laboratories (NREL) [9, 11]. Furthermore, a simplified model using Flex5 was built up for comparison to the general purpose multibody programs.

13.2 Theoretical Background

A short introduction into theoretical background of flexible multibody systems is given in this section. A detailed description can be found in [1] and in [16]. Flexible multibody systems are an extension to classical multibody systems. Classical multibody elements are consisting of rigid bodies connected via joints and force elements. An introduction into classical multibody systems is provided by [15] and [21]. To take elastic deformations into account, which are neglected by classical multibody systems, flexible multibody systems are developed. Flexible multibody systems consider typically mechanical systems with large rigid body motions and small deformations.

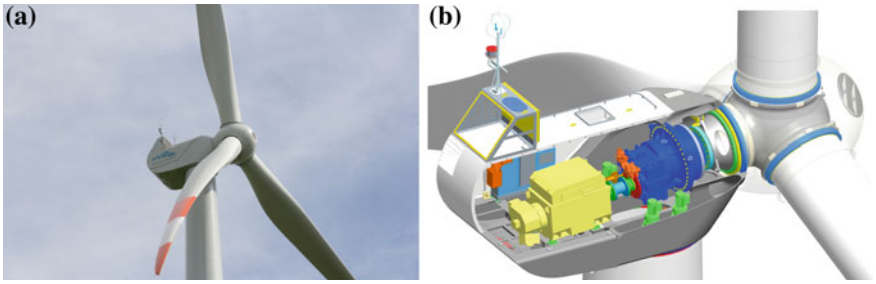


Fig. 13.1 Prototype of the 2.05 MW wind turbine erected in Tarnow, Mecklenburg-Western Pommerania. **a** Prototype. **b** CAD model of the drive train

13.2.1 Kinematics of Flexible Multibody Systems

The motion of rigid bodies in classical multibody systems is described by the translation and the rotation of the body-fixed reference frame. As it can be seen from Fig. 13.2, the deformation of a flexible body is described by a vectorial superposition of rigid body motions and elastic deformations, which is called floating frame of reference formulation.

Herein $\mathcal{K}_0$ with the origin O_0 represents a space-fixed reference frame and $\mathcal{K}_1$ with the origin O_1 the corresponding body-fixed reference frame. The relative position of the body-fixed reference frame $\mathcal{K}_1$ with respect to the space-fixed reference frame $\mathcal{K}_0$ is given by the vector $\mathbf{r}_{10}(t)$. The vector $\mathbf{r}_{21}(t)$ describes the undeformed position of an arbitrary point on the flexible body represented by the reference frame $\mathcal{K}_2$ relative to the body-fixed reference frame $\mathcal{K}_1$. The elastic deformation is described by the vector $\mathbf{u}_f(t)$. The overall position description of the reference frame $\mathcal{K}_3$ relative to the reference frame $\mathcal{K}_0$ is obtained by

$$ {}^0\mathbf{r}_{30}(t) = {}^0\mathbf{r}_{10}(t) + {}^{01}\mathbf{T}(t)\left({}^1\mathbf{r}_{21} + {}^1\mathbf{u}_f(t)\right). \tag{13.1} $$

The upper left index denotes the reference system in which the vectors are expressed. In general the elastic deformations ${}^1\mathbf{u}_f(t)$ have to be reformulated in order to reduce the degrees of freedom of the multibody system. Instead of a physical description of the elastic deformation in each point of the discretised body, global approaches, also called RITZ approaches, are used. Here, the physical deformation is described by a set of linear independent time-invariant global shape functions ${}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right)$ and corresponding time-dependent weighting factors representing the flexible coordinates of the multibody system ${}^1\mathbf{z}(t)$,

$$ {}^1\mathbf{u}_f(\mathbf{r}_{21}, t) = {}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right) {}^1\mathbf{z}(t). \tag{13.2} $$

The global shape functions in ${}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right)$ are obtained from a finite element analysis with a subsequent coordinate reduction by component mode synthesis. With

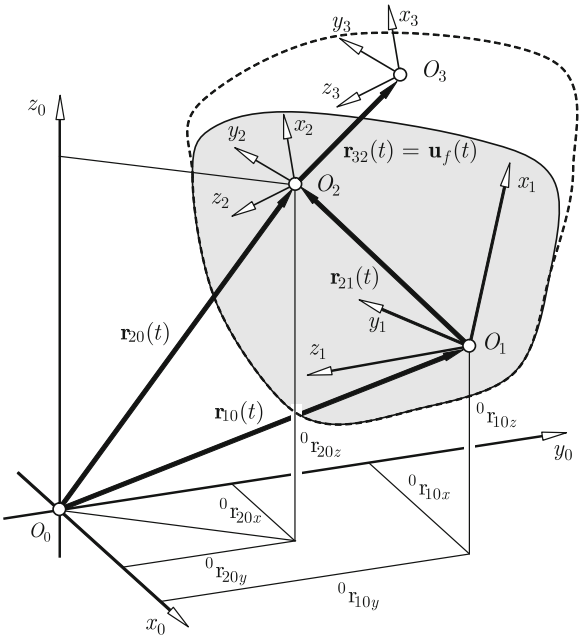


Fig. 13.2 Kinematics of a flexible body

(13.2) the position description of (13.1) can be rewritten as

$$ {}^0\mathbf{r}_{30}(t) = {}^0\mathbf{r}_{10}(t) + {}^{01}\mathbf{T}(t)\left({}^1\mathbf{r}_{21} + {}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right){}^1\mathbf{z}(t)\right). \tag{13.3} $$

Differentiation of (13.3) with respect to time leads to

$$ {}^0\dot{\mathbf{r}}_{30}(t) = {}^0\dot{\mathbf{r}}_{10}(t) + {}^{01}\dot{\mathbf{T}}(t)\left({}^1\mathbf{r}_{21} + {}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right){}^1\mathbf{z}(t)\right) + {}^{01}\mathbf{T}(t)\left({}^1\mathbf{Q}\left({}^1\mathbf{r}_{21}\right){}^1\dot{\mathbf{z}}\right). \tag{13.4} $$

Introducing the rotation parameters $\boldsymbol{\beta}$, for example EULER angles, BRYAN angles, or RODRIGUES parameters, the time derivative of the transformation matrix can be written as

$$ {}^{01}\dot{\mathbf{T}}(t) = \frac{\partial {}^{01}\mathbf{T}(t)}{\partial \boldsymbol{\beta}}\,\dot{\boldsymbol{\beta}}(t). \tag{13.5} $$

Then (13.4) can be written in matrix form by

$$ {}^0\dot{\mathbf{r}}_{30} = \left[\mathbf{E} \quad \frac{\partial {}^{01}\mathbf{T}(t)}{\partial \boldsymbol{\beta}} \left({}^1\mathbf{r}_{21} + {}^1\mathbf{Q} \left({}^1\mathbf{r}_{21} \right) {}^1\mathbf{z}(t) \right) \quad {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \left({}^1\mathbf{r}_{21} \right) \right] \begin{bmatrix} {}^0\dot{\mathbf{r}}_{10}(t) \\ \dot{\boldsymbol{\beta}}(t) \\ {}^1\dot{\mathbf{z}}(t) \end{bmatrix}, \tag{13.6} $$

with the identity matrix $\mathbf{E}$. As seen from (13.6), the motion of a flexible body is described by a translation and rotation of the body-fixed reference system $\mathcal{K}_1$ and the elastic deformations expressed in $\mathcal{K}_1$. The matrix on the right hand side of (13.6) is also written as

$$ {}^0\mathbf{L} = \left[\mathbf{E} \quad {}^0\mathbf{B} \left(\boldsymbol{\beta}, {}^1\mathbf{r}_{21}, \mathbf{z} \right) \quad {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \left({}^1\mathbf{r}_{21} \right) \right] \tag{13.7} $$

with the rotation matrix of the flexible body

$$ {}^0\mathbf{B}\left(\boldsymbol{\beta}, {}^1\mathbf{r}_{21}, \mathbf{z} \right) = \frac{\partial {}^{01}\mathbf{T}(t)}{\partial \boldsymbol{\beta}} \left({}^1\mathbf{r}_{21} + {}^1\mathbf{Q}\left({}^1\mathbf{r}_{21} \right) {}^1\mathbf{z}(t) \right). \tag{13.8} $$

The column vector on the right hand side of (13.6) represents the time derivatives of the coordinates of the flexible multibody system, which are summarised as

$$ {}^0\dot{\mathbf{q}} = \begin{bmatrix} {}^0\dot{\mathbf{r}}_{10}(t) \\ \dot{\boldsymbol{\beta}}(t) \\ {}^1\dot{\mathbf{z}}(t) \end{bmatrix}. \tag{13.9} $$

Thus, the overall motion is described as superposition of the rigid body translation and rotation of the body-fixed reference system in addition with the elastic deformations. Here, the rigid body motion is expressed with respect to the space-fixed reference system $\mathcal{K}_0$ whereas the elastic deformations are expressed with respect to the body-fixed reference system $\mathcal{K}_1$.

13.2.2 Equations of Motions of Flexible Multibody Systems

To describe the equations of motion of flexible multibody systems the LAGRANGE equations of the second kind are used,

$$ \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial T \left({}^0\mathbf{q}, {}^0\dot{\mathbf{q}}, t \right)}{\partial {}^0\dot{\mathbf{q}}^{\mathrm{T}}} \right) - \frac{\partial T \left({}^0\mathbf{q}, {}^0\dot{\mathbf{q}}, t \right)}{\partial {}^0\mathbf{q}^{\mathrm{T}}} = - \frac{\partial U \left({}^0\mathbf{q}, t \right)}{\partial {}^0\mathbf{q}^{\mathrm{T}}} + {}^0\mathbf{k}^{\mathrm{nk}}, \tag{13.10} $$

which are built up with respect to the space fixed reference system $\mathcal{K}_0$. Herein T represents the kinetic energy and U the potential energy of the mechanical system. The kinetic energy of a single body i is obtained by

$$T^i = \frac{1}{2} \int\limits_{V^i} \rho \, {}^0\dot{\mathbf{r}}_{30}^{\mathrm{T}} \, {}^0\dot{\mathbf{r}}_{30} \, \mathrm{d}V, \tag{13.11}$$

where ${}^0\dot{\mathbf{r}}_{30}$ is the velocity vector of an arbitrary point O_3 on the flexible body i with respect to the space-fixed reference system $\mathcal{K}_0$ and ρ is the density. Inserting (13.6) leads to

$$T^i = \frac{1}{2} \int\limits_{V^i} \rho \, {}^0\dot{\mathbf{q}}^{i\,\mathrm{T}} \, {}^0\mathbf{L}^{i\,\mathrm{T}} \, {}^0\mathbf{L}^i \, {}^0\dot{\mathbf{q}}^i \, \mathrm{d}V = \frac{1}{2} {}^0\dot{\mathbf{q}}^{i\,\mathrm{T}} \left(\int\limits_{V^i} \rho \, {}^0\mathbf{L}^{i\,\mathrm{T}} \, {}^0\mathbf{L}^i \, \mathrm{d}V \right) {}^0\dot{\mathbf{q}}^i. \tag{13.12}$$

The term in the squared brackets represents the mass matrix of body i, which can be expressed with (13.7) as

$${}^0\mathbf{M}^i = \int\limits_{V^i} \rho \, {}^0\mathbf{L}^{i\,\mathrm{T}} \, {}^0\mathbf{L}^i \, \mathrm{d}V = \int\limits_{V} \rho \begin{bmatrix} \mathbf{E} \\ {}^0\mathbf{B}^{\mathrm{T}} \\ \left({}^{01}\mathbf{T} \, {}^1\mathbf{Q}\right)^{\mathrm{T}} \end{bmatrix}^i \begin{bmatrix} \mathbf{E} & {}^0\mathbf{B} & {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \end{bmatrix}^i \mathrm{d}V. \tag{13.13}$$

The evaluation of the matrix product in (13.13) leads to

$${}^0\mathbf{M}^i = \int\limits_{V^i} \rho \begin{bmatrix} \mathbf{E} & {}^0\mathbf{B} & {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \\ & {}^0\mathbf{B}^{\mathrm{T}} {}^0\mathbf{B} & {}^0\mathbf{B}^{\mathrm{T}} \, {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \\ \mathrm{sym.} & & {}^1\mathbf{Q}^{\mathrm{T}} \, {}^1\mathbf{Q} \end{bmatrix}^i \mathrm{d}V = \begin{bmatrix} {}^0\mathbf{m}_{RR} & {}^0\mathbf{m}_{R\beta} & {}^{01}\mathbf{m}_{Rf} \\ & {}^0\mathbf{m}_{\beta\beta} & {}^{01}\mathbf{m}_{\beta f} \\ \mathrm{sym.} & & {}^1\mathbf{m}_{ff} \end{bmatrix}^i, \tag{13.14}$$

with the integrals

$$\begin{aligned} {}^0\mathbf{m}_{RR}^i &= \int\limits_{V^i} \rho \, \mathbf{E} \, \mathrm{d}V, & {}^0\mathbf{m}_{R\beta}^i &= \int\limits_{V^i} \rho \, {}^0\mathbf{B} \, \mathrm{d}V, & {}^{01}\mathbf{m}_{Rf} &= \int\limits_{V} \rho \, {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \, \mathrm{d}V, \\ {}^0\mathbf{m}_{\beta\beta} &= \int\limits_{V^i} \rho \, {}^0\mathbf{B}^{\mathrm{T}} {}^0\mathbf{B} \, \mathrm{d}V, & {}^{01}\mathbf{m}_{\beta f} &= \int\limits_{V^i} \rho \, {}^0\mathbf{B}^{\mathrm{T}} \, {}^{01}\mathbf{T} \, {}^1\mathbf{Q} \, \mathrm{d}V, & {}^1\mathbf{m}_{ff} &= \int\limits_{V^i} \rho \, {}^1\mathbf{Q}^{\mathrm{T}} \, {}^1\mathbf{Q} \, \mathrm{d}V. \end{aligned} \tag{13.15}$$

The overall kinetic energy of the flexible multibody system is obtained by the sum over the contributions of all bodies i,

$$T = \sum_i T^i = \bigcup_i \frac{1}{2} \, {}^0\dot{\mathbf{q}}^{i\,\mathrm{T}} \, {}^0\mathbf{M}^i \, {}^0\dot{\mathbf{q}}^i = \frac{1}{2} \, {}^0\dot{\mathbf{q}}^{\mathrm{T}} \, {}^0\mathbf{M} \, {}^0\dot{\mathbf{q}}. \tag{13.16}$$

The symbol $\bigcup\limits_i (\ldots)$ denotes the assembly of the mass matrices and coordinates over all bodies i of the multibody system. The derivatives of the kinetic energy T needed in (13.10) are obtained by

$$\frac{\partial T\left({}^0\mathbf{q},\,{}^0\dot{\mathbf{q}},\,t\right)}{\partial\,{}^0\dot{\mathbf{q}}^{\mathrm{T}}} = {}^0\mathbf{M}\,{}^0\dot{\mathbf{q}} \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T\left({}^0\mathbf{q},\,{}^0\dot{\mathbf{q}},\,t\right)}{\partial\,{}^0\dot{\mathbf{q}}^{\mathrm{T}}}\right) = {}^0\mathbf{M}\,{}^0\ddot{\mathbf{q}} + {}^0\dot{\mathbf{M}}\,{}^0\dot{\mathbf{q}}, \tag{13.17}$$

and

$$\frac{\partial T\left({}^0\mathbf{q},\,{}^0\dot{\mathbf{q}},\,t\right)}{\partial\,{}^0\mathbf{q}^{\mathrm{T}}} = \frac{\partial}{\partial\,{}^0\mathbf{q}^{\mathrm{T}}}\left(\frac{1}{2}\,{}^0\dot{\mathbf{q}}^{\mathrm{T}}\,{}^0\mathbf{M}\,{}^0\dot{\mathbf{q}}\right). \tag{13.18}$$

If bodies are flexible, they store potential energy in addition to the kinetic energy. The elastic potential energy U^i of a flexible body is equivalent to the deformation work applied to a flexible body. Hence, the elastic potential energy of single body i can be written as

$$U^i = \frac{1}{2}\int_{V^i} {}^1\boldsymbol{\varepsilon}^{\mathrm{T}}\,{}^1\boldsymbol{\sigma}\,\mathrm{d}V. \tag{13.19}$$

Here, $\boldsymbol{\sigma}$ represents the stress tensor and $\boldsymbol{\varepsilon}$ the strain tensor. By the use of the kinematic relationship between strain and displacements and by the use of the global Ritz approach of (13.2) the component vector of the strain tensor can be rewritten as

$${}^1\boldsymbol{\varepsilon} = {}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{u}_f = {}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\,{}^1\mathbf{z} \tag{13.20}$$

with the strain displacement relation matrix ${}^1\mathbf{D}_{\varepsilon u}$ represented in the body-fixed reference system $\mathcal{K}_1$. Assuming linear-elastic material behaviour and using (13.20) the stress tensor becomes

$${}^1\boldsymbol{\sigma} = {}^1\mathbf{C}\,{}^1\boldsymbol{\varepsilon} = {}^1\mathbf{C}\,{}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{u}_f = {}^1\mathbf{C}\,{}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\,{}^1\mathbf{z}. \tag{13.21}$$

Using (13.20) and (13.21) (13.19) can be rewritten as

$$\begin{aligned} U^i &= \frac{1}{2}\int_{V^i} {}^1\mathbf{z}^{\mathrm{T}}\left({}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\right)^{\mathrm{T}}\,{}^1\mathbf{C}\,{}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\,{}^1\mathbf{z}\,\mathrm{d}V \\ &= \frac{1}{2}\,{}^1\mathbf{z}^{\mathrm{T}}\left(\int_{V^i}\left({}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\right)^{\mathrm{T}}\,{}^1\mathbf{C}\,{}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\,\mathrm{d}V\right){}^1\mathbf{z}. \end{aligned} \tag{13.22}$$

Here, the term in the squared brackets represents the stiffness matrix of the flexible body i, which can be expressed as

$${}^1\mathbf{K}^i_{ff} = \int_{V^i}\left({}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\right)^{\mathrm{T}}\,{}^1\mathbf{C}\,{}^1\mathbf{D}_{\varepsilon u}\,{}^1\mathbf{Q}\,\mathrm{d}V. \tag{13.23}$$

As it can be seen from (13.22), the potential energy depends on the elastic displacements only. Using the coordinate vector of the flexible multibody system

$$ {}^0\mathbf{q}^i = \begin{bmatrix} {}^0\mathbf{r}_{10}(t) \\ \boldsymbol{\beta}(t) \\ {}^1\mathbf{z}(t) \end{bmatrix}^i , \tag{13.24} $$

(13.22) can be rewritten as

$$ U^i = \frac{1}{2} \begin{bmatrix} {}^0\mathbf{r}_{10} & \boldsymbol{\beta} & {}^1\mathbf{z} \end{bmatrix}^i \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^1\mathbf{K}^i_{ff} \end{bmatrix} \begin{bmatrix} {}^0\mathbf{r}_{10} \\ \boldsymbol{\beta} \\ {}^1\mathbf{z} \end{bmatrix}^i . \tag{13.25} $$

Finally, the overall elastic potential energy of the flexible multibody system is obtained by

$$ U = \sum_i U^i = \bigcup_i \frac{1}{2} {}^0\mathbf{q}^i \, {}^0\mathbf{K}^i \, {}^0\mathbf{q}^i = \frac{1}{2} {}^0\mathbf{q} \, {}^0\mathbf{K} \, {}^0\mathbf{q}. \tag{13.26} $$

The general non-conservative forces $\mathbf{k}^{\mathrm{nk}}$ in (13.10) can be expressed as

$$ {}^0\mathbf{k}^{\mathrm{nk}} = {}^0\mathbf{k}^{\mathrm{r}} + {}^0\mathbf{k}^{\mathrm{e}} = {}^0\mathbf{G}^{\mathrm{T}} \, {}^0\boldsymbol{\lambda} + {}^0\mathbf{k}^{\mathrm{e}}, \tag{13.27} $$

where $\mathbf{k}^{\mathrm{r}}$ and $\mathbf{k}^{\mathrm{e}}$ represents the general constraint and general applied forces, respectively. Typically the constraint forces are expressed as product of the constraint matrix $\mathbf{G}$ and the constraint force coordinates (LAGRANGE multipliers) $\boldsymbol{\lambda}$. Using (13.26) the derivative of the elastic potential energy U with respect the coordinates is obtained by

$$ \frac{\partial U \left({}^0\mathbf{q}, t \right)}{\partial {}^0\mathbf{q}^{\mathrm{T}}} = {}^0\mathbf{K} \, {}^0\mathbf{q}. \tag{13.28} $$

Inserting (13.17), (13.18), (13.27), and (13.28) into (13.10) leads to the equations of motion of a constrained flexible multibody system,

$$ {}^0\mathbf{M} \, {}^0\ddot{\mathbf{q}} - {}^0\mathbf{G}^{\mathrm{T}} \, {}^0\boldsymbol{\lambda} = {}^0\mathbf{K} \, {}^0\mathbf{q} + {}^0\mathbf{k}^{\mathrm{c}} + {}^0\mathbf{k}^{\mathrm{e}} \tag{13.29} $$

with the quadratic velocity term

$$ {}^0\mathbf{k}^{\mathrm{c}} = -{}^0\dot{\mathbf{M}} \, {}^0\dot{\mathbf{q}} + \frac{\partial}{\partial {}^0\mathbf{q}^{\mathrm{T}}} \left(\frac{1}{2} {}^0\dot{\mathbf{q}}^{\mathrm{T}} \, {}^0\mathbf{M} \, {}^0\dot{\mathbf{q}} \right) \tag{13.30} $$

containing the general CORIOLIS and gyroscopic forces. The equations of motion of a classical multibody system built up of rigid bodies can be obtained by assuming the flexible coordinates $\mathbf{z}$ and their time derivatives to be zero.

13.3 Control of Variable Speed-Variable Pitch-Horizontal Axis Wind Turbines

The scope of this chapter is to give a general introduction into control of horizontal axis wind turbines. Due to the high dynamics of wind turbines and their nonlinear behaviour, the control of the mechanical system is a very important aspect for the multibody simulation. The controller strategies can be divided into collective pitch control on the one hand and individual pitch control on the other hand. An overview on collective pitch control for wind turbines, representing classical controller strategies, can be found in [3] and [13]. Besides simple PI (proportional-integral) or PID (proportional-integral-derivative) controller algorithms, also optimal power tracking algorithms are treated. An introduction into individual pitch control for wind turbines, representing advanced controller schemes, is given by [4]. The advanced controller strategies mainly focus on load reduction of the wind turbine. An extensive overview of advanced controller schemes including active tower damping procedures and controller schemes with wind prediction systems such as light detection and ranging (LIDAR) systems is given within the UPWIND project, see [6]. A field test validation of the advanced controller schemes on two- and three-bladed wind turbines is given by [5].

For introduction into control of a horizontal axis wind turbine, a closer look into the correlation of aerodynamic inflow and mechanical power of the wind turbine rotor is necessary. According to [8] the power of the aerodynamic inflow is given by

$$P_{\text{Wind}} = \frac{1}{2}\,\rho\,\pi\,r_{\text{Rotor}}^2\,u_{\text{Wind}}^3, \tag{13.31}$$

where ρ is the density of air, r_{Rotor} is the radius of the wind turbine rotor and u_{Wind} is the velocity of the aerodynamic inflow or short wind speed. The mechanical power of the rotor is obtained by multiplying (13.31) with a power coefficient c_P,

$$P_{\text{Rotor}} = \frac{1}{2}\,\rho\,\pi\,r_{\text{Rotor}}^2\,u_{\text{Wind}}^3\,c_P(\vartheta,\lambda), \tag{13.32}$$

which is a function of the of the pitch angle ϑ and the tip speed ratio λ. The tip speed ratio λ can be calculated by

$$\lambda = \frac{\omega_{\text{Rotor}}\,r_{\text{Rotor}}}{u_{\text{Wind}}} = \frac{\omega_{\text{Gen}}\,r_{\text{Rotor}}}{i_{\text{Gear}}\,u_{\text{Wind}}}, \tag{13.33}$$

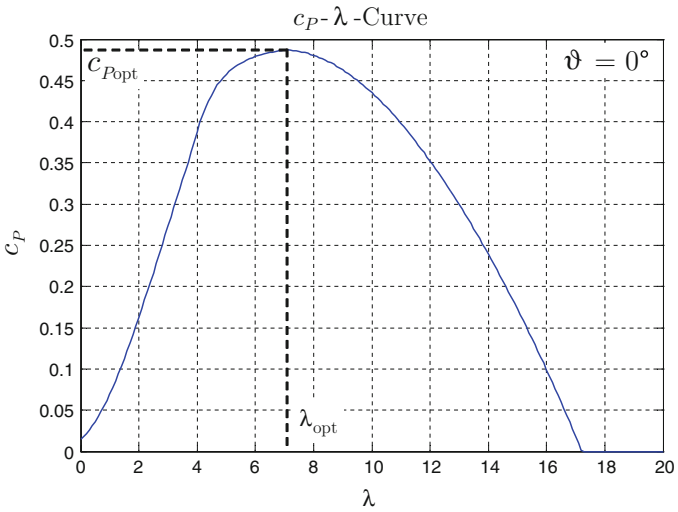


Fig. 13.3 Typical c_P-λ-curve of a horizontal axis wind turbine

with the angular velocity of the rotor ω_{Rotor}, the angular velocity of the generator ω_{Gen} and the gear ratio i_{Gear}. The theoretical maximum power which can be gripped by the rotor from the aerodynamic inflow is given by the BETZ limit, see [2], with a value of $c_{P\max} = 16/27 \approx 0.59$. The technical maximum power coefficient c_P lies around 0.5, this means the rotor converts around 50% of the energy of aerodynamic inflow into mechanical energy. A typical c_P-λ-curve of a horizontal axis wind turbine for pitch angle $\vartheta = 0°$ is shown in Fig. 13.3.

The optimal tip speed ratio λ_{opt} of a horizontal axis wind turbine lies between 7 and 9. The aim of the controller is to keep the wind turbine at an operating state with maximum earnings. Therefore, the wind turbine operates most time around the point $\left(c_{P\text{opt}}, \lambda_{\text{opt}}\right)$. Because the inflow velocity varies over the time, the optimal power output of the wind turbine is ensured by controlling the speed of the wind turbine. The control principle of a variable speed-variable pitch-horizontal axis wind turbine is shown in Fig. 13.4.

The blue curves represent the mechanical power achieved at the rotor at certain wind speeds with a pitch angle $\vartheta = 0°$. The wind speed u_{Wind} is varied from 4 m/s to 12 m/s in steps of 1 m/s. These curves are calculated by (13.32), where the power coefficient $c_P(\vartheta, \lambda)$ has to be varied over the angular velocity of the generator ω_{Gen} due to its dependence on the tip speed ratio λ according to (13.33). If (13.33) is resolved with respect to the inflow velocity u_{Wind} and inserted into (13.32), the power at the rotor can be obtained by

$$P_{\text{Rotor}} = \frac{1}{2}\,\rho\,\pi\,r_{\text{Rotor}}^5\,\frac{c_P(\vartheta,\lambda)}{\lambda^3}\left(\frac{\omega_{\text{Gen}}}{i_{\text{Gear}}}\right)^3, \tag{13.34}$$

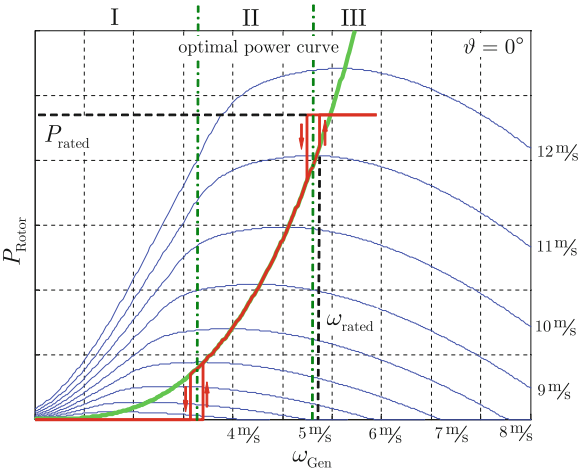


Fig. 13.4 Control principle of a horizontal axis wind turbine

Using the optimal power coefficient c_{Popt} and the optimal tip speed ratio λ_{opt} from Fig. 13.3 in (13.34) the optimal power curve, represented by the green curve in Fig. 13.4, is obtained. The optimal power curve is a function of the angular velocity of the generator only, $P_{\text{opt}} \sim \omega_{\text{Gen}}^3$. The red curve in Fig. 13.4 represents the behaviour of a wind turbine using classical control schemes. The operating states of a wind turbine can be divided into the sections I, II and III. Within section I the power output of the wind turbine is zero, representing the start up of the wind turbine. At the end of section I the prescribed cut-in generator angular velocity is reached. The wind turbine operates on the optimal power curve in section II below the rated angular velocity of the generator ω_{rated}. Within this section the angular velocity of the wind turbine is variable and controlled by the generator torque. At the end of section II the wind turbine is controlled off the optimal power curve in order to reduce the tip speed of the blades for reasons of noise. Within section III the controller keeps the generator angular velocity at almost constant level around the rated generator angular velocity ω_{rated} by pitching the blades and thus decreasing the power output of the rotor. Different generator set points are used at the transition of the sections II and III to avoid the different controller schemes from interfering each other. At the transition of the sections I and II the same principle is applied to avoid a choke off of the wind turbine during start up. This prevents jumps in the generator torque leading to dynamical impacts in the drive train of the wind turbine.

As mentioned before different control schemes are developed for wind turbines. PI or PID controller schemes are widely used in wind turbines. These controller schemes are a good starting point for most horizontal axis wind turbines. A PID controller can be written as

$$y = \left(K_p + \frac{K_i}{s} + \frac{K_d\, s}{1 + s\, \tau} \right) x, \tag{13.35}$$

with the LAPLACE variable s. Herein, the control error and the actuating signal are represented by x and y, respectively. K_p, K_i, and K_d are the proportional, integral and differential gains. To avoid the differential term becoming large at high frequencies the time constant τ is introduced. A PI controller is obtained by setting the differential gain K_d to zero.

For section II, thus below rated wind speed, the input error signal x in the PID controller is given by the difference between the actual angular velocity and the optimal angular velocity of the generator provided by the optimal power curve. The angular velocity is controlled by the generator torque representing the control action y. In section III, thus above rated wind speed, the input error signal x is given by the difference between the actual angular velocity of the generator ω_{Gen} and the rated angular velocity of the generator ω_{rated}. In this case the actuating signal is the pitch velocity.

13.4 The Multibody Model and its Interaction with the Aerodynamic Code and the Controller

The aim of this section is to describe the multibody model and its interaction to other codes which are necessary to simulate an overall wind turbine model. First, the interaction of the multibody model with the aerodynamic code and the controller is presented. Then, a general description of the different simulation models follows.

13.4.1 Interaction Scheme of the Multibody Program with the Aerodynamic Code and the Controller

The simplified interaction scheme of the multibody simulation with the aerodynamic code and the controller of the wind turbine is shown in Fig. 13.5.

A discrete interface was developed for the interaction of the controller with the multibody program. The aim of this interface is to integrate the same controller software into the multibody simulation as implemented on the physical wind turbine. The interaction scheme represents a software-in-the-loop principle and was developed in analogy to the hardware-in-the-loop principle described in [22]. The multibody program contacts the controller at discrete time steps and waits until the controller provides the corresponding output data. Due to the fact that the integrator of the multibody program generally has a variable step size, the interface has to be realised in such a way that the controller is contacted only at prescribed constant time steps. The controller on the physical wind turbine operates with a time step of 10 ms, which is also chosen as prescribed time step for the interface. Between the discrete time steps the output values of the previous time steps are used and kept

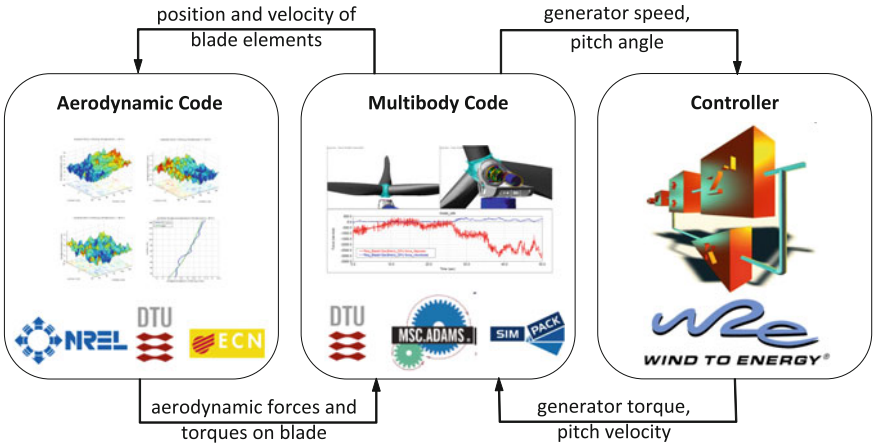
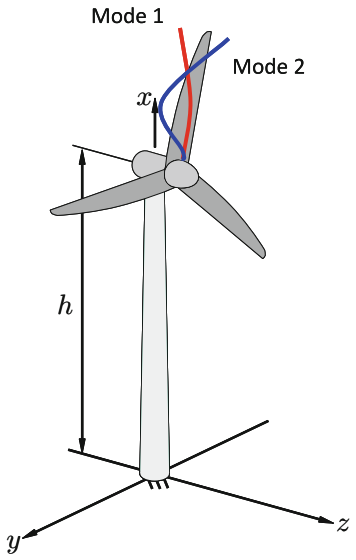


Fig. 13.5 Simplified interaction scheme of the wind turbine model

constant. This interface scheme does not present any real-time capabilities which is, however, not necessary and not realisable for large models .

As mentioned before, the wind turbine is pitch-controlled and has a variable speed. As shown in Fig. 13.5, the main input values of the controller are the generator speed and the pitch angle of the blades, depending to the operating mode according to Fig. 13.4. The corresponding actuating signals of the controller are the angular pitch velocity and the generator torque, which are applied to the multibody model as imposed motions, represented by consistent positions, velocities and accelerations, and applied forces, respectively. Indeed, there are several additional values which are transferred to the controller and given back to the multibody program that are not described here because of their large number. The aims of these values are, for example, to ensure electric power quality or decrease vibrations of the drive train. As the simulation is not limited by any real-time requirements, the in- and output values of the controller is not limited as long as the programmable logic controller on the physical wind turbine has enough capabilities to calculate the controller program in real-time.

The multibody model also interacts with the aerodynamic code. The aerodynamic code used in Flex5 was developed at Danish Technical University and is integrated directly into the multibody code. The aerodynamic code used in SIMPACK and Adams is based on AeroDyn provided by the National Renewable Energy Laboratories (NREL). Both aerodynamic codes include a calculation based on the blade element momentum theory. Therefore, the blade is divided into separate aerodynamic elements. As shown in Fig. 13.5, the multibody code provides the position and velocity of the blade elements. The aerodynamic code provides the aerodynamic forces and moments. In addition, the aerodynamic code from NREL is extended by the general dynamic wake theory which is based on the acceleration potential theory, see [18]. Furthermore, SIMPACK provides an interface to the aerodynamic code



Mode	Description
1-6	Tower Foundation
7-8	Tower bending thrust direction
9-10	Tower bending side-side
11	Tower top yaw
12	Tower top tilt
13	Rotation of main shaft
14	Bending of main shaft x-axis
15	Bending of main shaft y-axis
16	1st flapwise mode of blade 1
17	2nd flapwise mode of blade 1
18	1st edgewise mode of blade 1
19	2nd edgewise mode of blade 1
20	1st flapwise mode of blade 2
21	2nd flapwise mode of blade 2
22	1st edgewise mode of blade 2
23	2nd edgewise mode of blade 2
24	1st flapwise mode of blade 3
25	2nd flapwise mode of blade 3
26	1st edgewise mode of blade 3
27	2nd edgewise mode of blade 3
28	Torsion of main shaft

Fig. 13.6 Degrees of freedom of the Flex5 wind turbine model

from ECN based on the lifting line theory, see [20]. For comparison of the results of the multibody codes used, the blade element momentum theory is applied within this research work only.

13.4.2 Simple Multibody Model of the Wind Turbine in Flex5

The Flex5 model has a fixed topology comprising overall 28 degrees of freedom as shown in Fig. 13.6.

Within this given topology, the model of a specific wind turbine is implemented by defining a fixed set of parameters. A general parameter file contains, above all, the geometric parameters, the load case parameters, a tower parameter file, and two parameter files for the blade containing the mechanical and aerodynamic properties. An implementation of another topology of the model is possible by a modification of the computer code only.

The elasticity of the blade and the tower is represented by a superposition of the first two modes in the two independent directions, respectively. The degrees of freedom can be switched off independently within the general parameter file. Due to the small number of degrees of freedom, the calculation time is very short. A typical run of a ten minute time series with turbulent wind conditions on an *Intel Core i7-2600* takes about one minute.

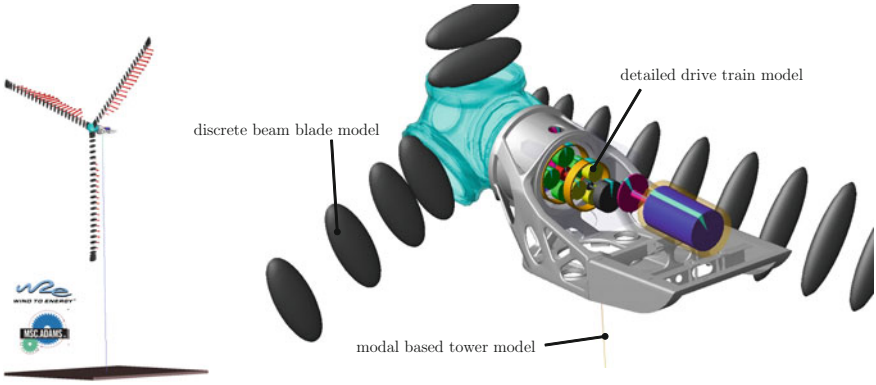


Fig. 13.7 Multibody model of the wind turbine using MSC.Adams

13.4.3 Parametric Multibody Model of the Wind Turbine in MSC.Adams

The multibody model in MSC.Adams is parametrically built up. That means, instead of defining the model within the Adams/View preprocessing environment, the model is created within the MATLAB environment. The MATLAB code generates an Adams command file in the ASCII format, which can be imported by Adams/View. The same principles were also applied for model generation of the high-lift mechanisms of a modern transport aircraft, see [23].

The Adams model comprises a flexible tower model based on a finite element model, blades built up of discrete beams and a detailed drive train model, see [14]. The discrete beams consist of lumped mass elements and EULER–BERNOULLI beams. Compared to a blade model consisting of flexible bodies, a higher numerical stability of discrete beams in MSC.Adams during start-up of the wind turbine could be achieved. Furthermore, effects like centrifugal stiffness are taken into account, and the interface loads along the blade can be obtained easily.

As a result, a multibody model with approximately 600 degrees of freedom is obtained, see Fig. 13.7. The higher model depth of the simulation leads to larger CPU times compared to Flex5. The simulation of a ten minute time series with turbulent wind conditions on an *Intel Core i7-2600* takes about 20 minutes.

13.4.4 Flexible Multibody Model of the Wind Turbine in SIMPACK

The detailed multibody model developed in SIMPACK 8.903b is described in [14]. For the results represented within this chapter, a revised model of the 2.05 MW wind turbine developed in SIMPACK 9.3 is used.

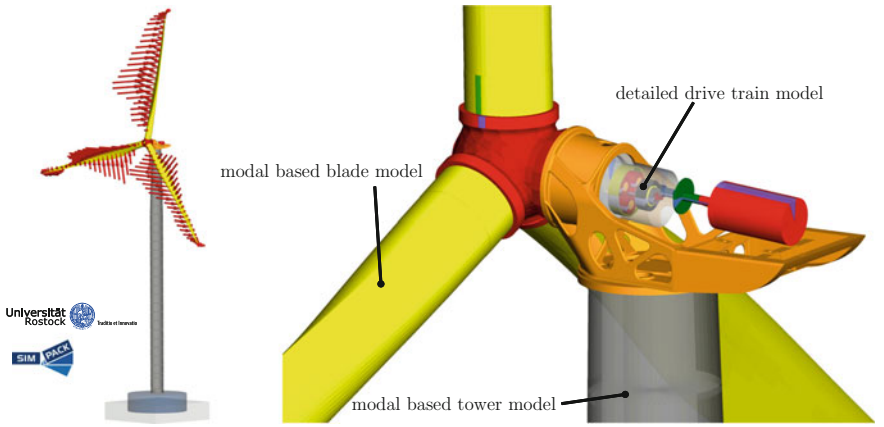


Fig. 13.8 Multibody model of the wind turbine using SIMPACK

For comparison the tower and the blades are modelled elastically. A detailed drive train model including two planetary gear stages and a spur gear stage is integrated, comprising torsional stiffness of shafts and the contact-stiffness of interacting gears. Blades, tower, and drive train are substructures within the main model and therefore are easy to exchange in order to simulate different wind turbine designs. In order to gain comparable results in accordance to measurements at the wind turbine in the field, the original controller is implemented using an interface. The flexible bodies are based on a finite element formulation. The tower consists of solids and TIMOSHENKO beam elements. Additionally lumped masses are used. The blades are modelled by TIMOSHENKO beam elements. Modal superposition techniques are used to reduce the number of degrees of freedom of the finite element models and to speed up simulation time. As a result, the wind turbine model consists of 44 degrees of freedom, see Fig. 13.8.

For simulating different load cases time efficiently, an open source code script developed by SIMPACK and refined for the 2.05 MW wind turbine is used. A ten minute time series with turbulent wind conditions takes approximately 15 minutes on an *Intel Core i5-2320*.

13.5 Comparison of Simulations and Measurements

A very important aspect is the experimental validation of the simulations which are also needed for type certification of the wind turbine. Therefore, the prototype of the wind turbine is equipped with numerous measurement sensors, e.g. strain gauges at the blade root, in the tower or at the low speed shaft, see Fig. 13.9a. In addition, a measurement mast equipped with wind vanes and cup anemometers is built up in front of the wind turbine, see Fig. 13.9b.

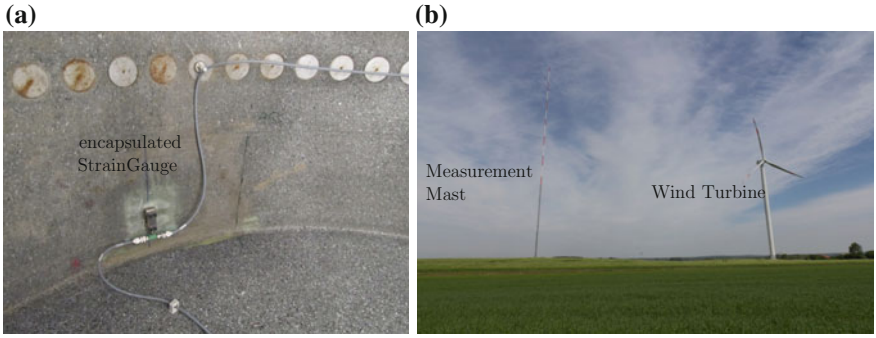


Fig. 13.9 Measurement setup of the wind turbine. **a** Strain gauge at the blade root. **b** Prototype of the 2.05 MW wind turbine and measurement mast

The aim of this research work is not to establish one of the simulation environments as a reference but to compare equivalent models developed by means of the simulation packages with the measurement results. The objective is to evaluate the different modelling concepts used in the packages for wind turbine simulation. In contrast to this method, other research projects validate their program development for wind turbine simulation by comparison to generally accepted design codes, see [19]. The differences in model generation and build-up of the equations of motion leads to difficulties in the direct comparison of the design codes. In the author's opinion, it is more meaningful to evaluate each program by comparing the numerical results with real measurements. Hence, the scatter plots in the next sections compare the simulations of Flex5, MSC.Adams and SIMPACK with measurements from the prototype of the wind turbine.

The continuous time series obtained from the measurements are split up into ten minute time series in analogy to the simulated time series. The resulting time series are classified with respect to mean wind speed and the turbulence intensity of the wind. To compare simulations and measurements, statistical evaluations of the calculated and measured results are done, as it is difficult to transfer the wind conditions from the measurement to the simulations. The statistical values used for comparison are the minimum, maximum, mean value and standard deviation. All calculations and measurements are done with a turbulence intensity of 10 %. For statistical confidence, the calculations are done with different wind seeds. A comparison of all measured interface loads and operating values is not possible within this chapter. For comparison of the wind turbine behaviour the measured and calculated electrical power, pitch angle and rotor speed are chosen. To evaluate the simulated loads the bending moments at blade root and the tilt bending moment at tower base are compared. The vertical green line within the statistical diagrams denotes the rated wind speeds. Because the predominant loads are applied on the blades, the bending moments at the blade root are chosen for a comparison of the dynamic loads using rainflow counting procedures, see [10]. The load cycles are estimated for the product life cycle with respect to the wind distribution in the wind class GL IIa according to the GL guideline [7].

13.5.1 Electrical Power (Power Curve)

The power curve is important for the economics of a wind turbine. High earnings especially in the part-load operational range leads to a fast return of invest, which is requested by the operators of wind farms. The maximum, minimum, mean electrical power and the corresponding standard deviations of the Flex5, MSC.Adams and SIMPACK simulations compared to the measurement results are shown in Fig. 13.10.

The diagrams show a very good agreement of simulations with Flex5 and MSC.Adams compared measurements. The simulation results (maximum, mean value and standard deviation) using SIMPACK lie below the measured curves, but show in general a good agreement. The use of the same controller on the prototype and within the simulations contributes to this result.

13.5.2 Pitch Angle

The pitch angle shows the working principle of the controller, especially of the pitch controller above rated wind speed. The maximum, minimum, mean pitch angle of blade 1 and the corresponding standard deviations of the Flex5, MSC.Adams and SIMPACK simulations compared to the measurement results are shown in Fig. 13.11.

The diagrams show a very good agreement of simulations with Flex5 and SIMPACK compared to measurements. The simulation results (maximum and mean value) using MSC.Adams lie above the measured curves. It can be noticed that the slope of the simulated mean values is a little bit steeper compared to measurements. The use of the same controller on the prototype and within the simulations contributes to the good agreement of simulation and measurements.

13.5.3 Rotor Speed

Another method to check the working principle of the controller is the comparison of the rotor speed, which can be calculated by $n_{\mathrm{Rotor}} = 30\,\omega_{\mathrm{Rotor}}/\pi$. The maximum, minimum, mean rotor speed and the corresponding standard deviations of the Flex5, MSC.Adams and SIMPACK simulations compared to the measurement results are shown in Fig. 13.12.

As can be seen in Fig. 13.12 the mean values of simulation and measurement agree very well around rated wind speed. Below rated wind speed the simulation results using SIMPACK lie below the measurements. The same behaviour has already been noticed for the comparison of simulated and measured electrical power. Furthermore the maximum values around rated wind speed lie above the measured values using MSC.Adams and SIMPACK. For the rotor speed the best agreement between simulation and measurement are reached using the Flex5 code.

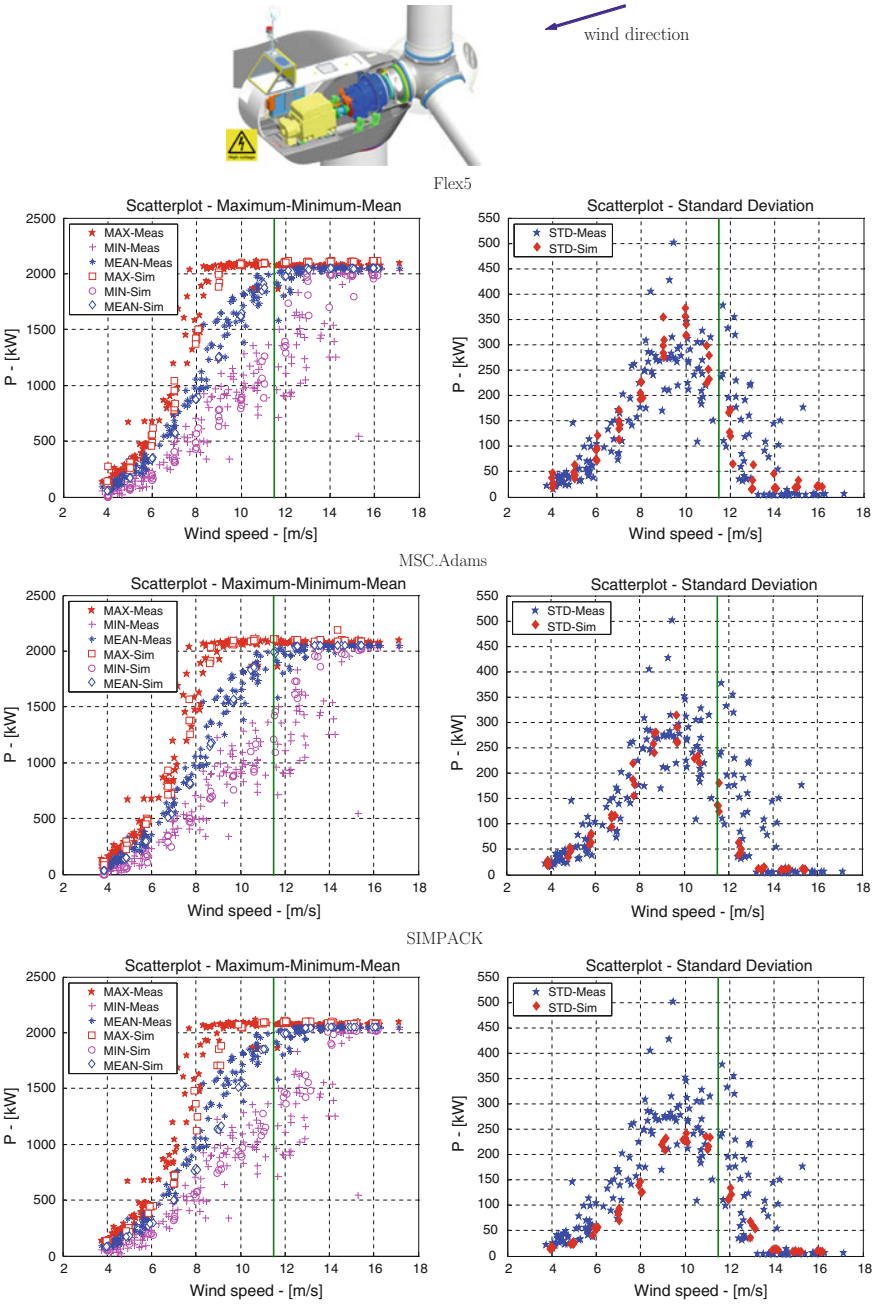


Fig. 13.10 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the electrical power

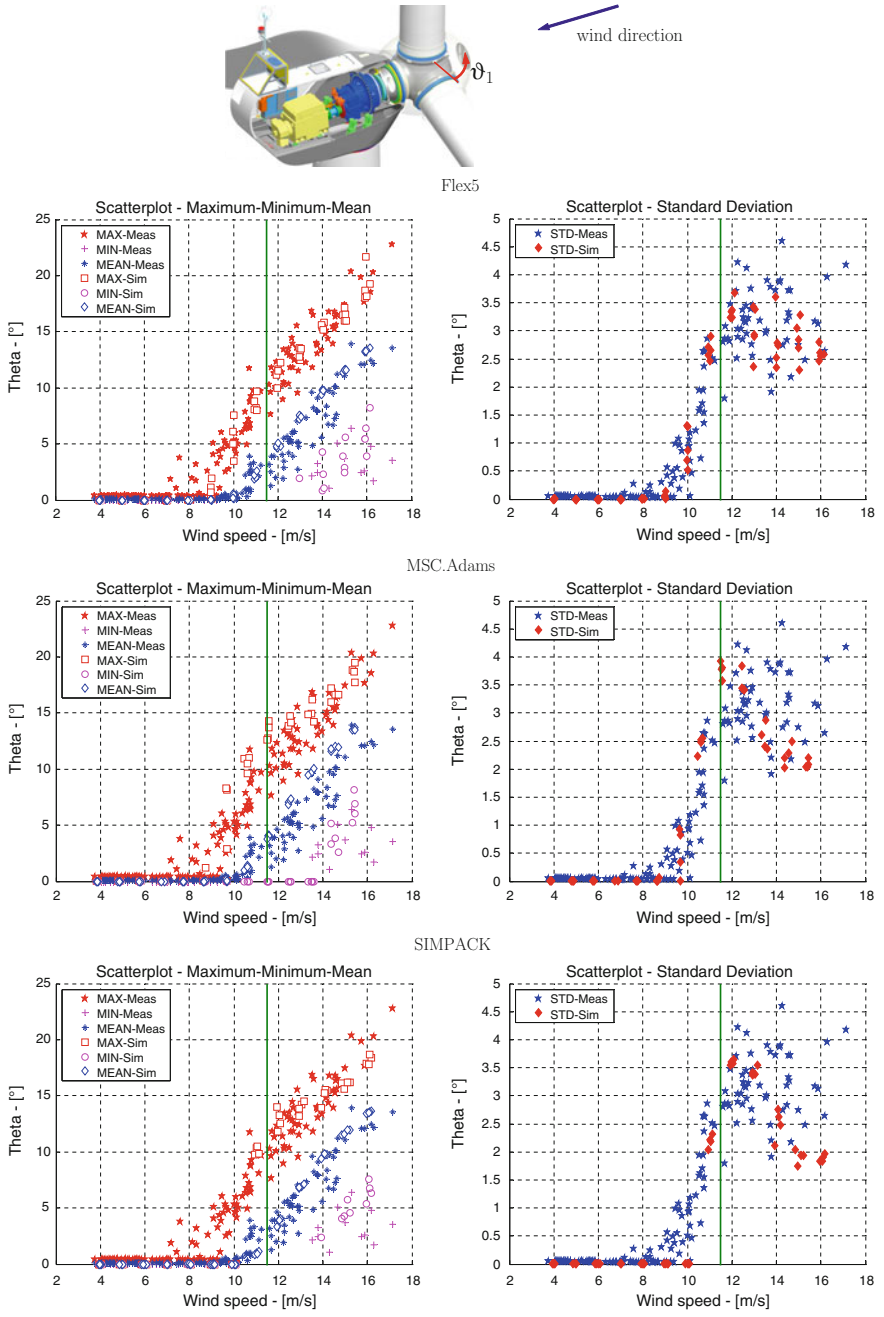


Fig. 13.11 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the pitch angle ϑ_1

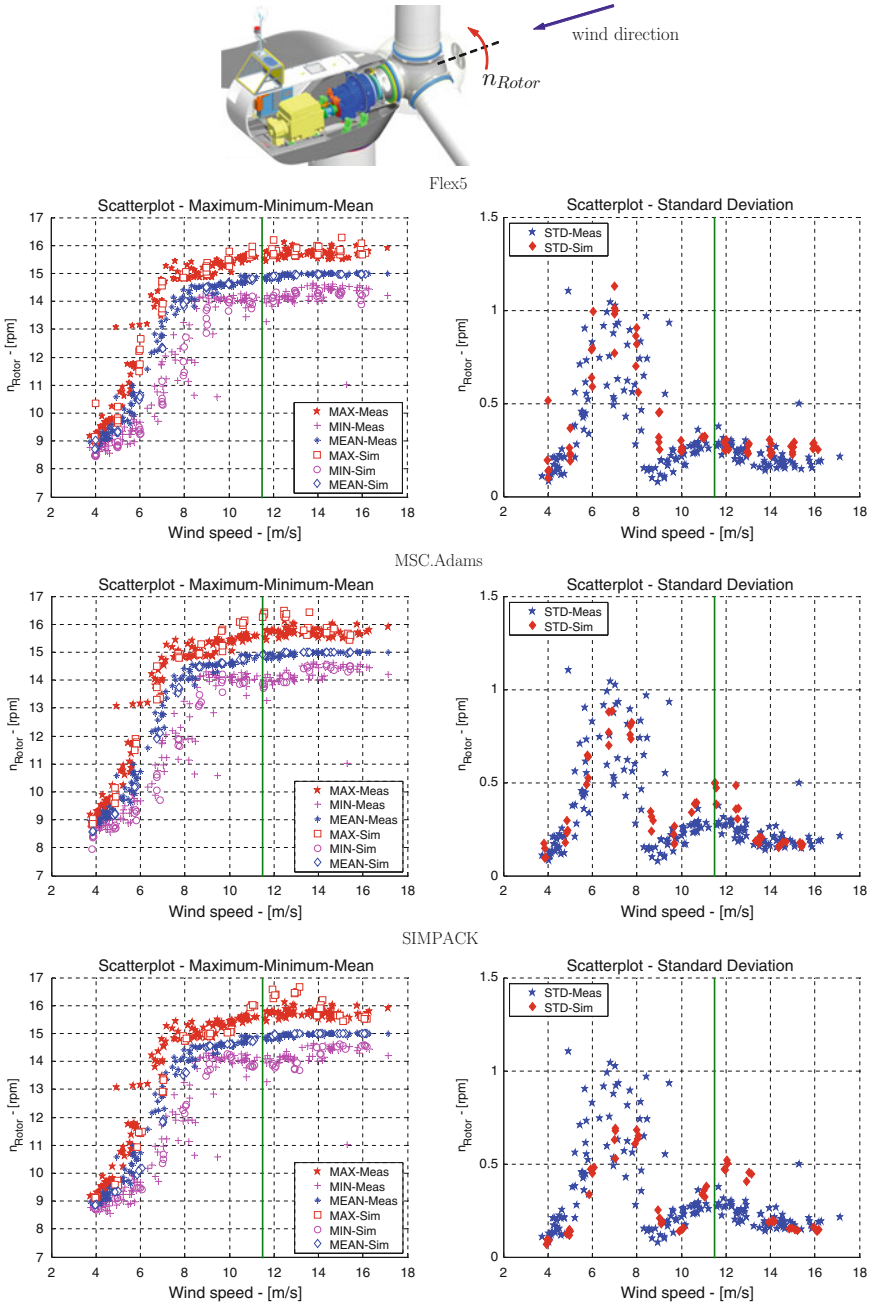


Fig. 13.12 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: Comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the rotor speed

13.5.4 Flapwise Bending Moment at the Blade Root

The flapwise bending moment at the blade root is caused by the lift of the aerodynamic profile of the blade. The maximum, minimum, and mean flapwise bending moment at the blade root as well as the corresponding standard deviation of the Flex5, MSC.Adams and SIMPACK simulations compared to the measurement results are shown in Fig. 13.13.

The direction of the moment M_{flap} and the wind direction in Fig. 13.13 indicates that the values of the bending moment are negative. The mean values obtained from the simulations with MSC.Adams und SIMPACK with the measurements show a better agreement compared to the Flex5 simulations. Especially, this is seen at rated wind speed, that means around 10–11 m/s. However, the absolute values from the Flex5 simulations exceed those from the measurements, indicating that the Flex5 simulations lead to conservative load estimations.

Beside the statistical evaluation of flapwise bending moment at the blade root a dynamic evaluation has been done. Therefore the rainflow matrix is estimated from the time series of the flapwise bending moment. The flapwise bending moments are strongly influenced by the turbulence of the wind. The rainflow matrices of the simulations are presented in Fig. 13.14.

From the rainflow matrix show a large number of load cycles at a small load range at any mean values. The corresponding rainflow matrix of the measurements is presented in Fig. 13.15.

Due to the stochastic behaviour of the wind during measurements and the use of different wind simulators during the simulations a qualitative comparison of the rainflow matrices is difficult. Also a quantitative comparison is not an appropriate way of comparison because of the discrete classification during the rainflow counting procedure. To circumvent these problems the load cycles are summed over each load range class neglecting their mean values. Subsequently, the load cycles are accumulated in that way that load cycles with a large load range comprises all load cycles with smaller load ranges. The corresponding diagrams of the load range vs. accumulated load cycles are presented in Fig. 13.14. As can be seen the simulations using Flex5 show a non-conservative behaviour for large load ranges compared to measurements. In contrast the comparison of simulations using MSC.Adams and SIMPACK and measurements show a good agreement or conservative behaviour for dynamic loads over the whole accumulated load cycle range.

13.5.5 Edgewise Bending Moment at the Blade Root

The edgewise bending moment at the blade root is mainly caused by the dead weight of the blade. Furthermore, the loads of the dead weight of the blade are superposed by dynamical mass effects and aerodynamic loads due to the lift of the blade. The maximum, minimum, and mean edgewise bending moment at the blade root and the corresponding standard deviation of the Flex5, MSC.Adams and SIMPACK simulations compared to measurement results are shown in Fig. 13.16.

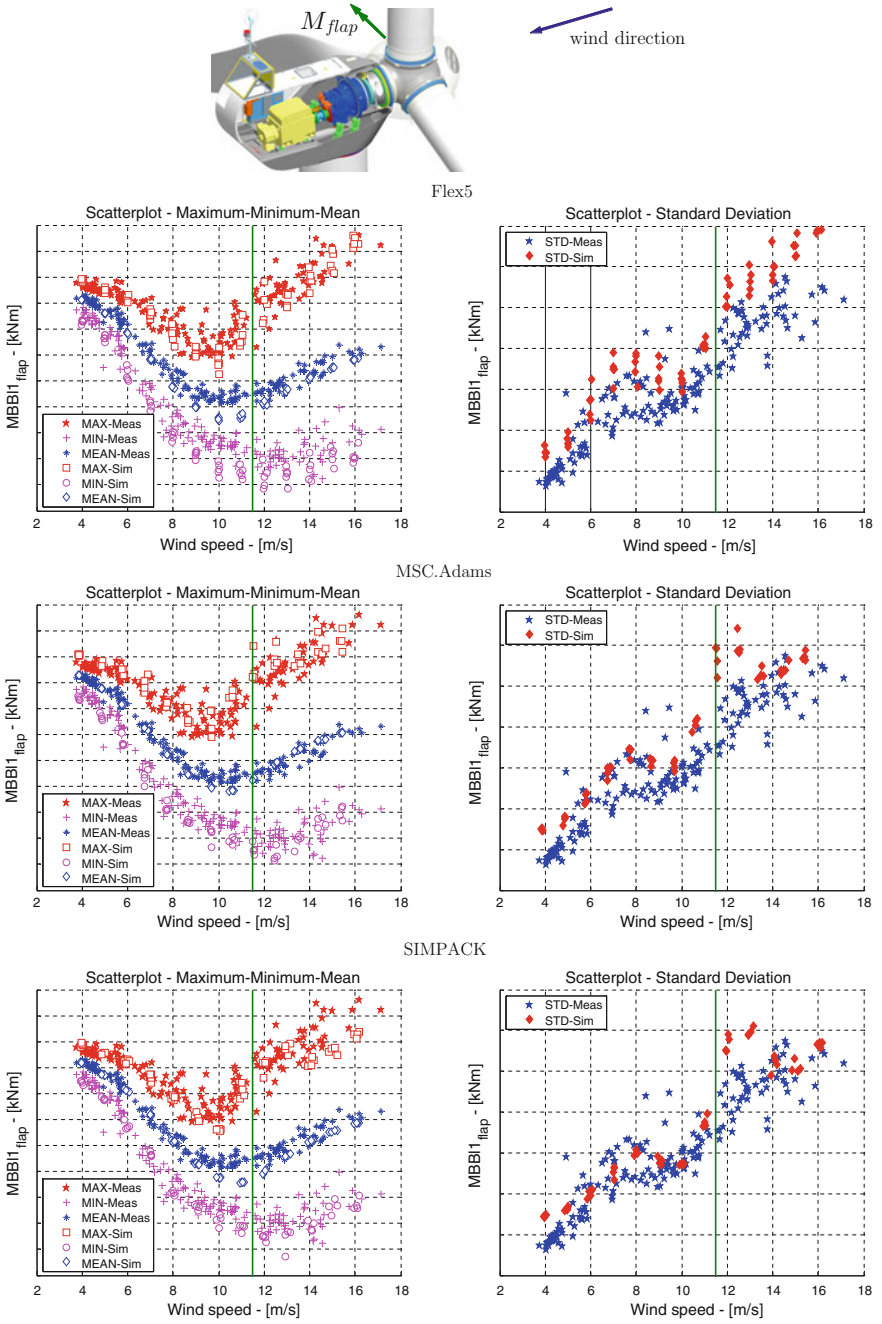


Fig. 13.13 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the flapwise bending moment at blade root

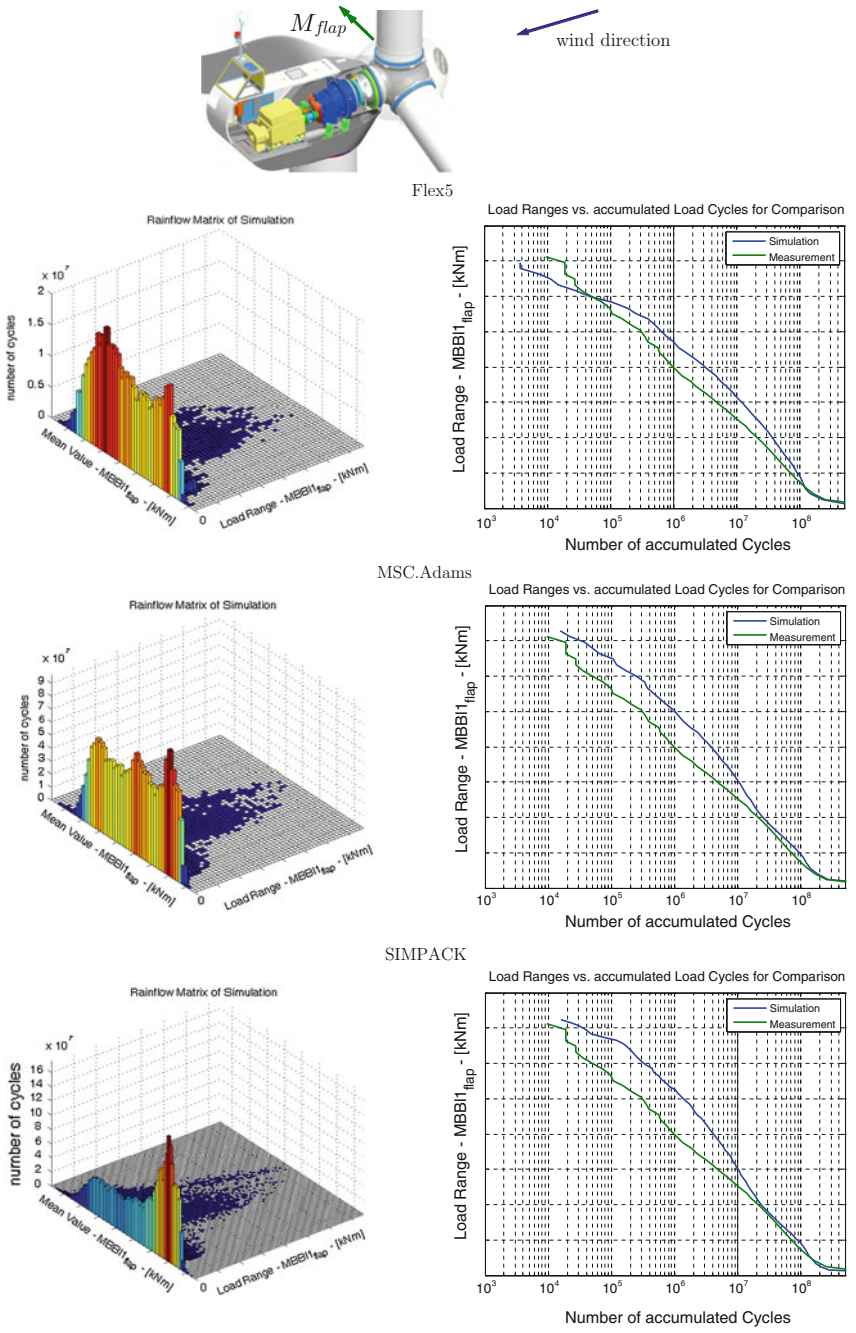


Fig. 13.14 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: comparison of the dynamic values (rainflow matrix, load range vs. accumulated load cycles) of the flapwise bending moment at the blade root

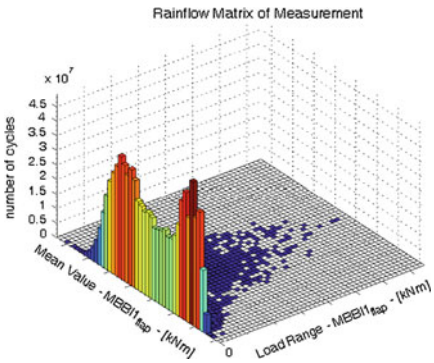


Fig. 13.15 Rainflow matrix of measurement: flapwise bending moment at the blade root

The comparison of simulations and measurement show some differences in the statistical values. A possible uncertainty is the calibration of the strain gauges at the blade root. The sensors are calibrated by the dead weight and a slow revolution of the wind turbine. Typically, the wind turbine coasts freely during calibration. The comparison of the standard deviation shows the smallest differences between the simulation of the blade model in MSC.Adams and measurement. The simulation with SIMPACK shows a good agreement of minimum, maximum and mean value, but large differences in the standard deviation. Thus, the blade model used in SIMPACK should be modified to reproduce the characteristics of the real blade.

Due to the strong influence of the dead weight the rainflow matrix has a typical characteristic. Comparison of the rainflow matrices of the simulations in Fig. 13.17 and the rainflow matrix of the measurement in Fig. 13.18 show three peaks within all diagrams. The single peak can be approximated by the first order static moment of the blade and the revolutions of the wind turbine during its life cycle.

However, the comparison of the load range vs. the accumulated load cycles in Fig. 13.17 show a conservative behaviour of all simulations for edgewise bending moment compared to measurements. It can be denoted from the step in the diagrams that large loads acts on the wind turbine with more than $1 \cdot 10^8$ load cycles during its product life cycle.

13.5.6 Tilt Bending Moment at the Tower Base

The tilt bending moment at the tower base is mainly caused by the thrust of the rotor. Also some dynamic effects of the mass of the nacelle and the rotor affects the tilt bending moment at the tower base. The maximum, minimum, and mean tilt bending moment at the tower base and the corresponding standard deviation of the Flex5, MSC.Adams and SIMPACK simulations compared to measurement results are shown in Fig. 13.19.

Due to the fact that the wind turbine almost behaves like a cantilever beam with a single mass at its end all simulation packages provide a very good agreement compared to measurement. The large influence of the rotor thrust results in same characteristics of flapwise bending moment and tilt bending moment at tower base.

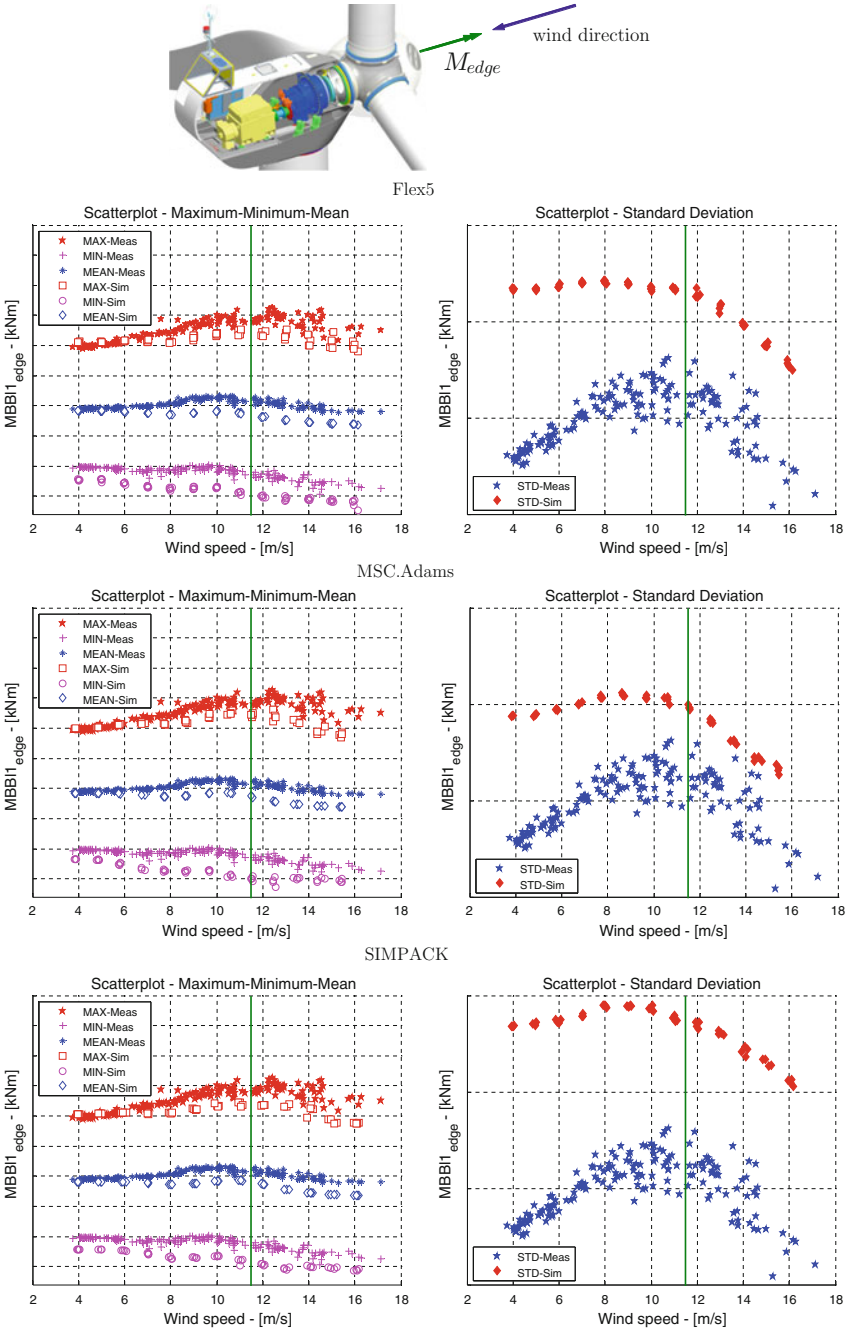
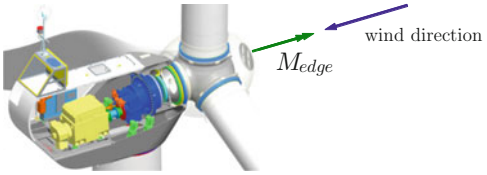
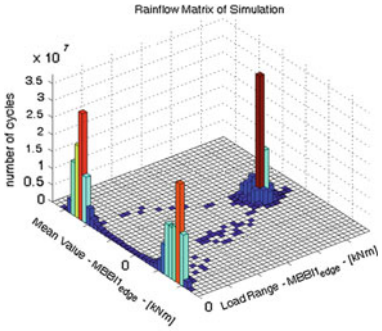
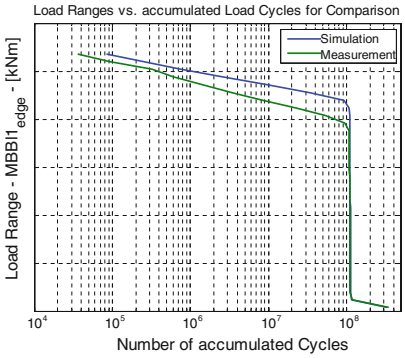


Fig. 13.16 Measurement and simulation with Flex5, MSC.Adams and SIMPACK: comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the edgewise bending moment at the blade root

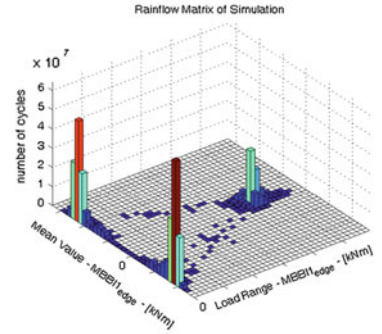


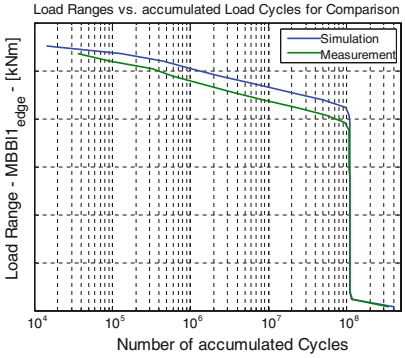
Flex5



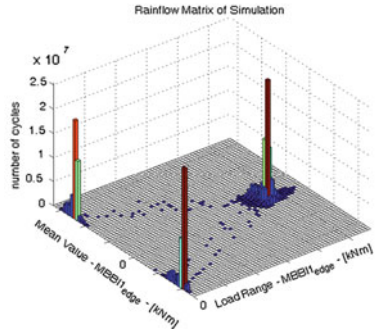


MSC.Adams





SIMPACK



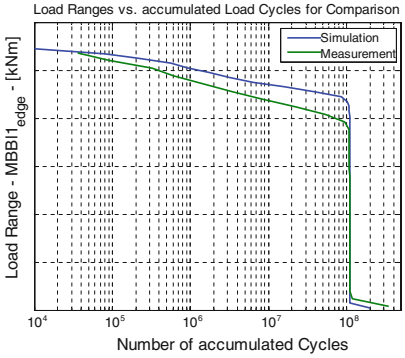


Fig. 13.17 Measurement and simulations with Flex5, MSC.Adams and SIMPACK: comparison of the dynamic values (rainflow matrix, load range vs. accumulated load cycles) of the edgewise bending moment at the blade root

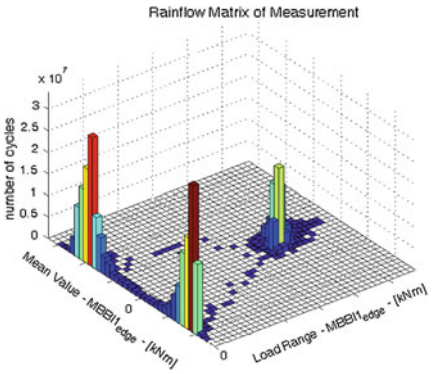


Fig. 13.18 Rainflow matrix of measurement: edgewise bending moment at the blade root

13.6 Conclusion

The present contribution compares different multibody codes for simulation of a variable speed-variable pitch-horizontal axis wind turbine. The wind turbine simulation tool Flex5 and the general purpose multibody codes MSC.Adams and SIMPACK are used for comparison. The same controller schemes as used on the real wind turbine are implemented into the simulation environments. A prototype of the 2.05 MW wind turbine were erected and a lot of measurements was done within a period of 1.5 years. The wind turbine was simulated using turbulent wind models. For statistical confidence, different wind seeds are used in the different simulation environments. For the predominant loads, the bending moments at the blade root, a dynamical evaluation of the time series has been done.

The comparison shows that in general all tested simulation tools are suitable for load calculation of wind turbines. The use of the same controller as on the physical wind turbine leads to a similar wind turbine behaviour in simulation and field test as it can be seen in the comparison of the electrical power, the pitch angle, and the rotor speed. However, the more detailed MSC.Adams and SIMPACK models show a better agreement of simulation and measurement for the bending moment at blade root, especially in flapwise direction. A general purpose multibody program enables to use more degrees of freedom for wind turbine modelling compared to special wind turbine design codes. However, the more complex models are more prone to modelling or parameter errors. Especially the model of the blades has to be regarded carefully. But also the wind turbine simulation code Flex5 has its advantages because of the short calculation times compared to the general purpose multibody codes. For example, if a test of different blade designs has been done in the preliminary design phase, then a qualitative comparison is possible using Flex5.

The simulation models are validated for typical power production load cases. The validated models can also be used for the calculation of extreme load cases, for example gust load cases. The use of detailed multibody models could lead to more lightweight designs of the wind turbines, reducing its costs and increasing its efficiency. Furthermore, a more detailed multibody model leads to an increase of

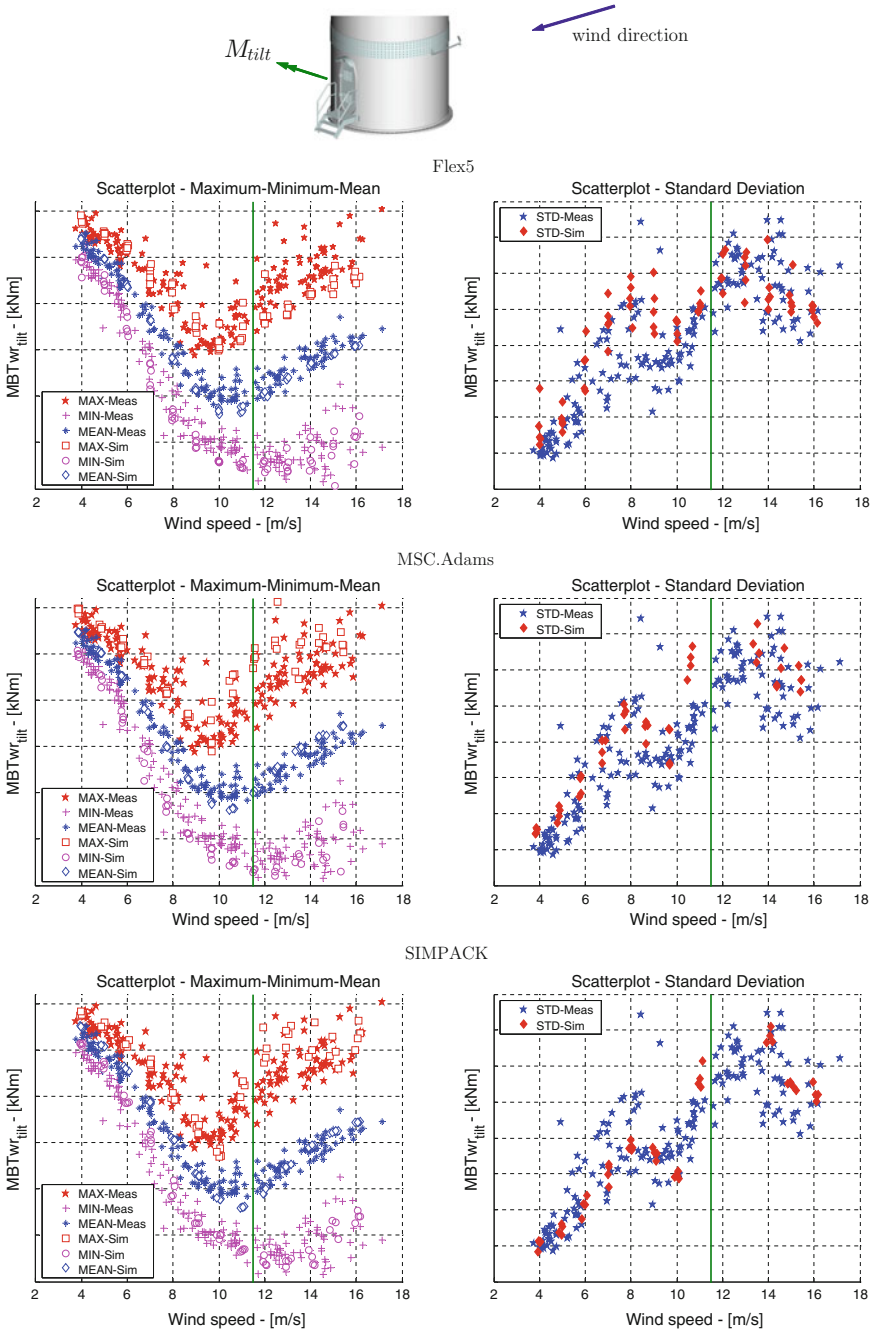


Fig. 13.19 Measurement and simulation with Flex5, MSC.Adams and SIMPACK: comparison of the statistical values (maximum, minimum, mean value, standard deviation) of the tilt bending moment at the tower base

the understanding of the dynamically behaviour of the wind turbine and provide the possibility to any arbitrary extension of the wind turbine model such as tower vibration absorbers.

Acknowledgment The authors would like to thank the German Federal Ministry for the Environment, Nature Conservation and Nuclear Safety.

References

1. Bauchau OA (2010) Flexible multibody dynamics. Springer, New York
2. Betz A (1926) Wind-Energie und ihre Ausnutzung durch Windmühlen. Aerodynamische Versuchsanstalt, Göttingen
3. Bossanyi EA (2000) The design of closed loop controllers for wind turbines. Wind Energy 3:149–163
4. Bossanyi EA (2003) Individual blade pitch control for load reduction. Wind Energy 6:119–128
5. Bossanyi EA, Fleming PA, Wright AD (2013) Validation of individual pitch control by field tests on two- and three-bladed wind turbines. IEEE Trans Control Syst Technol 21(4):1067–1078
6. Bossanyi EA, Savini B, Iribas M, Hau M, Fischer B, Schlipf D, van Engelen T, Rossetti M, Carcangui CE (2012) Advanced controller research for multi-MW wind turbines in the UPWIND project. Wind Energy 15:119–145
7. Germanischer Lloyd (2010) Guideline for the certification of wind turbines edition 2010. Germanischer Lloyd, Hamburg
8. Hau E (2010) Wind turbines: fundamentals, technologies, application economics. Springer, New York
9. Jonkman, BJ, Jonkman JM (2013) Addendum to the user's guides for FAST, A2AD, and AeroDyn released March 2010-February 2013. Tech rep, National Renewable Energy Laboratory, Golden, Colorado.
10. Köhler M, Jenne S, Pötter K, Zenner H (2012) Zählverfahren und Lastannahme in der Betriebsfestigkeit, German edn. Springer, Berlin
11. Laino DJ, Hansen AC (2002) AeroDyn v12.50: user's guide. Tech rep, Windward Engineering (prepared for the National Renewable Energy Laboratory), Salt Lake, City.
12. MSC. Software (2012) Adams 2012.2: Adams 2012 Online help tech. rep., MSC Software Corporation, Santa Ana, California.
13. Munteanu I, Bratcu AI, Cutululis NA, Ceanga E (2008) Optimal control of wind energy systems: towards a global approach (advances in industrial control). Springer, New York
14. Rachholz R, Woernle C, Zierath J (2012) Dynamics of a controlled flexible multi-body model of a 2 MW wind turbine. In: Proceedings of the 2nd joint international conference on multibody system dynamics-IMSD, Stuttgart.
15. Schiehlen W, Eberhard P (2004) Technische Dynamik. Teubner B.G, GmbH, Leipzig
16. Shabana AA (2010) Dynamics of multibody systems. Cambridge University Press, Cambridge
17. Simpack: Simpack v9.3, (2013) Documentation to Simpack. Tech rep, Simpack AG, Gilching, Germany
18. Suzuki A (2000) Application of dynamic inflow theory to wind turbine rotors. Ph.D. thesis, Department of Mechanical Engineering, University of Utah, Salt Lake City.
19. Taubert M, Clauß S, Freudenberg H, Keil A, März M, Moser W, Wulf HO (2011) Wind Turbine Design Codes: Eine Validierung von alaska/Wind mit BLADED, FAST und FLEX5. Tech rep Institut für Mechatronik, Chemnitz
20. van Garrel A (2003) Development of a wind turbine aerodynamics simulation module. Tech. rep, ECN Wind Energy, Petten, The Netherlands

21. Woernle C (2011) Mehrkörpersysteme: eine Einführung in die Kinematik und Dynamik von systemen starrer Körper, German edn. Springer, Berlin
22. Woernle C, Kaehler M, Rachholz R, Herrmann S, Zierath J, Souffrant R, Bader R (2010) Robot-based HiL test of joint endoprostheses. In: Lenarcic J (ed) Advances in robot kinematics: analysis and control. Springer, Berlin
23. Zierath J, Woernle C, Heyden T (2009) Elastic multibody models of transport aircraft high-lift mechanisms. AIAA J Aircraft 46(5):1513–1524

Chapter 14
A Real-Time Multibody Dynamics Model for an Unmanned Robot Vehicle Based on the Subsystem Synthesis Method

Myoung-Ho Kim, Hee Chan Kang and Sung-Soo Kim

Abstract In this chapter, real-time multibody dynamics models for an unmanned robot vehicle have been developed. The unmanned robot vehicle consists of six identical suspension subsystems. The suspension system comprises an MR-rotary damper and air springs with a double slider-crank mechanism. A 1/6 robot vehicle model was constructed, and then a full vehicle model was effectively generated by synthesizing six 1/6 robot vehicle models, using the subsystem synthesis method. An explicit–implicit integrator has been employed for the stable solutions. In order to improve efficiency, a model with simplified suspensions was also developed. The simplified suspension model consists of just a rotational spring-damper. Equivalent spring characteristics were obtained from approximating the characteristics of the original model. Through the rough terrain run simulations, the computational efficiency of the subsystem synthesis method was investigated with regard to the formulations, subsystem model simplification, and different integration methods.

14.1 Introduction

Unmanned robot vehicles have been developed for military applications, including for surveillance, mine removing, and even assault purposes [6, 21]. Most of the military purpose unmanned robot vehicles contain several wheel-suspension subsystems

M.-H. Kim · H. C. Kang
Graduate School of Mechanical Design and Mechatronics Engineering, Chungnam National University, 220 Kung-dong, Yusong-Ku, Daejeon 305-764, Republic of Korea
e-mail: mh_kim@cnu.ac.kr

H. C. Kang
e-mail: hc_kang@cnu.ac.kr; hs_yun@cnu.ac.kr

S.-S. Kim (✉)
Department of Mechatronics Engineering, Chungnam National University, 220 Kung-dong, Yusong-Ku, Daejeon 305-764, Republic of Korea
e-mail: sookim@cnu.ac.kr

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_14,

Fig. 14.1 Prototype of unmanned robot vehicle

for high-speed operation in off-road situations. Figure 14.1 shows the prototype of the unmanned robot vehicle developed by the Korean Agency of Defense Development. It is equipped with several different types of sensors, such as 3D laser scanners, vision cameras, and radars, in order to gather information for autonomous and remote operations. Real-time dynamics models for unmanned robot vehicles are becoming important for the following applications. The first application is Hardware-In-the Loop Simulations (HILS) for the controller development of the suspension subsystems for the robot vehicle. HILS simulations are very effective at developing and validating the suspension control logics. The real-time dynamics model is essential for HILS, since the actual suspension control hardware system is interfaced with the software model of the unmanned robot vehicle system. The responses of the software model must be computed much faster than the actual hardware controller loop time, considering the overhead from the interfacing between the hardware system and software model [17]. The second application is associated with a training simulator for the operators [19]. Most of the unmanned robot vehicles are not only autonomously operated but also remotely controlled. The maneuvering of robot vehicles is highly dependent on the skill of the operators. A training simulator is necessary to improve the operators' remote control ability. The training simulator basically consists of the operator's console with a monitoring system for displaying the virtual scenes, as well as a force feedback human interface, a control loading system, a real-time dynamics model, and a virtual reality graphic display system. In this simulator, depending on the manipulation of the human interface, the response from the dynamics model of the unmanned robot vehicle must be produced much faster than the real-time response due to the overhead from the virtual reality graphic computation. The last application for the real-time dynamics model is on-board simulation to emulate various sensors. As an example of virtual accelerometer to determine the optimal speed of the unmanned robot vehicle before actually moving forward, the real-time on-board simulations must be carried out to

predict vertical accelerations based on the scanned terrain data from the 3D laser scanners, which are located on the front of the vehicle. Based on the predicted vertical acceleration, the optimal forward speed can be determined [14]. The vertical acceleration must be computed much faster than the real-time, due to the overhead from the scanned data processing.

Real-time formulations for the multibody systems were developed in the early 80s in the field of robotics. Walker and Orin [25] developed the efficient Order N^3 formula. Featherstone [8, 9] developed a fully recursive Order N formulation for open chain systems in which the computational amount of the algorithms increases linearly as the number of bodies increases in the open chain. Bae and Huag [2, 3] generalized Featherstone's idea for the tree topology system and also extended the closed-loop systems. Bae et al. [4] improved the Order N^3 formulation when there was a small number of bodies in the chain. Also, Tsai and Haug [24] applied this improved algorithm to the vehicle system for real-time application with a driving simulator. The subsystem synthesis method has been developed by mixing the Order N^3 formulation and the embedded procedure from the Order N formulation by Kim [15]. This method has proved to be ideally suited for a system with several identical subsystems attached to the main body.

The subsystem synthesis method has been applied to a vehicle system with several identical suspension systems in the real-time application. The joint coordinate has been used in the method with the explicit integrator for the HILS application [17]. For the stiff suspension subsystems, the subsystem synthesis method, based on the Cartesian coordinate, was developed with the implicit integrator [16]. However, in this case, it is difficult to achieve real-time simulations without efficient linear equation solvers. In order to overcome the limitation arising from the Cartesian coordinate formulation, the joint coordinate subsystem synthesis method was developed with an explicit–implicit integrator [18].

One of the advantages of the subsystem synthesis method is that it provides a modular structure for the program, since the equations of motion for each suspension subsystem can be generated separately, while the equations of motion for the chassis can also be constructed separately. The coupling effects between each of the subsystems and the chassis are considered by the effective mass matrices and the effective force vectors. As a result of these characteristics of the subsystem synthesis method, suspension models with different degrees of complexity can be easily implemented using a library of suspension subsystems. Furthermore, different integration methods can be applied to the subsystem equations of motion and the chassis equations of motion. The explicit–implicit integrator with the subsystem synthesis method was developed by applying an explicit integrator to the chassis' equations of motion and by employing an implicit integrator to the subsystem equations of motion [10, 12, 13, 18].

In this chapter, real-time multibody dynamics models for the unmanned robot vehicle have been developed by using the above described features of the subsystem synthesis method. The unmanned robot vehicle consists of six identical suspension subsystems. Each of the suspension system is composed of an MR-rotary damper and air springs with a double slider-crank mechanism. The joint coordinate subsystem

synthesis method has been applied to this model to achieve the real-time simulations. The performance of the subsystem synthesis method has been compared with that of the conventional method with joint coordinates. In order to improve efficiency, a model with simplified suspensions was also developed. The simplified suspension system consists of a load arm with a rotational spring-damper. The equivalent spring characteristics were obtained from the original model. Using rough terrain run simulations, the two different models described previously were compared in order to investigate the solution accuracy and efficiency. The explicit–implicit integration method was applied to improve the efficiency with a larger stepsize. Finally, the performance of the real-time models was investigated with regard to the different formulations, different suspension modeling (i.e., model simplification), and different integration schemes.

14.2 Real-Time Model Using the Subsystem Synthesis Method

This section presents the use of the subsystem synthesis method to create equations of motion for the unmanned robot vehicle. The procedure for how to synthesize the several subsystem equations of motion to the equations of motion for the chassis body will be explained without detailed derivations of subsystem equations of motion. The detailed derivations have been presented in Ref. [15].

14.2.1 A 1/6 Robot Vehicle Model

Figure 14.2 represents a 1/6 robot vehicle model in which the suspension subsystem consists of an MR-rotary damper and air springs with a double slider-crank mechanism. The figure on the right in Fig. 14.2 shows a schematic diagram of the model. The edge denotes a joint, and the circle represents a body in the multibody system. The crank is connected to the chassis with a revolute joint. The crank is also attached to two pistons through two connecting rods, which are modeled as massless links. Each piston is connected to the housing with a translational joint.

The system equations of motion for this 1/6 robot vehicle model can be derived using the recursive kinematics and variational approach [15]. The virtual base body of the subsystem is introduced as the reference body in the subsystem for the kinematics with relative joint coordinates. In other words, the virtual base body is the base reference body in the recursive kinematics [15]. The wheel body is connected to the housing with a fixed joint, since the simple tire model is used, which does not consider wheel spin dynamics and only includes vertical stiffness of the tire. The fixed joint connects the virtual base body and the chassis in order to make a dynamically equivalent system.

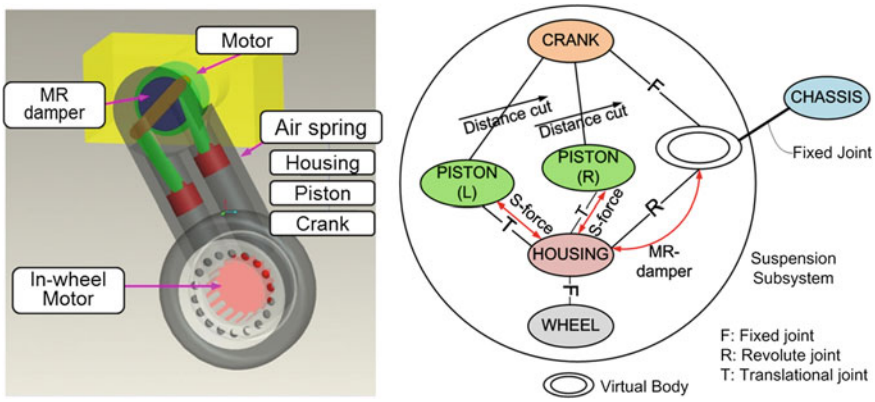


Fig. 14.2 A 1/6 robot vehicle model

If the conventional joint coordinate formulation [24] is used, the equations of motion for the suspension subsystems with the virtual base body can be derived as:

$$\begin{bmatrix} \bar{\mathbf{M}}_{yy} & \bar{\mathbf{M}}_{yq} & \mathbf{0} \\ \bar{\mathbf{M}}_{yq}^T & \bar{\mathbf{M}}_{qq} & \boldsymbol{\Phi}_{\bar{\mathbf{q}}}^T \\ \mathbf{0} & \boldsymbol{\Phi}_{\bar{\mathbf{q}}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\hat{\mathbf{Y}}}_0 \\ \ddot{\bar{\mathbf{q}}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{P}}_y \\ \bar{\mathbf{P}}_q \\ \bar{\boldsymbol{\gamma}} \end{bmatrix} \tag{14.1}$$

where $\dot{\hat{\mathbf{Y}}}_0$ is the chassis' state acceleration vector, $\ddot{\bar{\mathbf{q}}}$ is the joint acceleration vector associated with the suspension subsystem, and $\boldsymbol{\lambda}$ is the Lagrange multiplier vector associated with the distance constraints that represent the massless connecting rods of the double slider-crank mechanism. $\bar{\mathbf{M}}_{yy}$, $\bar{\mathbf{M}}_{yq}$, and $\bar{\mathbf{M}}_{qq}$ are the inertia matrices associated with the suspension subsystem and $\bar{\mathbf{P}}_y$ and $\bar{\mathbf{P}}_q$ are the generalized force vectors. $\boldsymbol{\Phi}_{\bar{\mathbf{q}}}$ is the Jacobian matrix associated with two distance constraints. The state acceleration vector $\dot{\hat{\mathbf{Y}}}_0$ has dimensions of six, and the joint acceleration vector has dimensions of three (one for the revolute joint between the chassis and the housing and two for the two translational joints). The Lagrange multipliers vector $\boldsymbol{\lambda}$ has dimensions of two.

From the second and third rows of Eq. (14.1), the equations of motion for the subsystem can be obtained by treating the state acceleration $\dot{\hat{\mathbf{Y}}}_0$ of the chassis body as the known value:

$$\begin{bmatrix} \bar{\mathbf{M}}_{\mathbf{qq}} & \boldsymbol{\Phi}_{\bar{\mathbf{q}}}^{\mathbf{T}} \\ \boldsymbol{\Phi}_{\bar{\mathbf{q}}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\mathbf{q}}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{P}}_{\mathbf{q}} - \mathbf{M}_{\mathbf{yq}}^{\mathbf{T}} \dot{\hat{\mathbf{Y}}}_{\mathbf{0}} \\ \bar{\boldsymbol{\gamma}} \end{bmatrix} \tag{14.2}$$

Then solving for $\boldsymbol{\lambda}$ yields:

$$\boldsymbol{\lambda} = \left(\boldsymbol{\Phi}_{\bar{\mathbf{q}}} \bar{\mathbf{M}}_{qq}^{-1} \boldsymbol{\Phi}_{\bar{\mathbf{q}}}^T \right)^{-1} \left\{ \boldsymbol{\Phi}_{\bar{\mathbf{q}}} \bar{\mathbf{M}}_{qq}^{-1} \left(\bar{\mathbf{P}}_q - \bar{\mathbf{M}}_{yq}^T \dot{\hat{\mathbf{Y}}}_0 \right) - \bar{\boldsymbol{\gamma}} \right\} \tag{14.3}$$

From the 2nd row of Eq. (14.1), the equation of motion can be expressed as:

$$\bar{\mathbf{M}}_{qq}\ddot{\bar{\mathbf{q}}} = \bar{\mathbf{P}}_q - \mathbf{M}_{yq}^T\dot{\hat{\mathbf{Y}}}_0 - \mathbf{\Phi}_{\bar{\mathbf{q}}}^T\boldsymbol{\lambda} \tag{14.4}$$

After obtaining the acceleration expression $\ddot{\bar{\mathbf{q}}}$ in terms of $\dot{\hat{\mathbf{Y}}}_0$ from Eq. (14.4) and substituting this expression and the expression of the Lagrange multiplier in Eq. (14.3) into the first row of Eq. (14.1), the reduced form of the virtual base body equations of motion can be obtained as

$$\breve{\mathbf{M}}^c\dot{\hat{\mathbf{Y}}}_0 = \breve{\mathbf{P}}^c \tag{14.5}$$

where,

$$\breve{\mathbf{M}}^c = \bar{\mathbf{M}}_{yy} - \bar{\mathbf{M}}_{yq}\bar{\mathbf{M}}_{qq}^{-1}\bar{\mathbf{M}}_{yq}^T + \bar{\mathbf{M}}_{yq}\bar{\mathbf{M}}_{qq}^{-1}\mathbf{\Phi}_{\bar{\mathbf{q}}}^T(\mathbf{\Phi}_{\bar{\mathbf{q}}}\bar{\mathbf{M}}_{qq}^{-1}\mathbf{\Phi}_{\bar{\mathbf{q}}}^T)^{-1}\mathbf{\Phi}_{\bar{\mathbf{q}}}\bar{\mathbf{M}}_{qq}^{-1}\bar{\mathbf{M}}_{yq}^T \tag{14.6}$$

$$\breve{\mathbf{P}}^c = \bar{\mathbf{P}}_y - \bar{\mathbf{M}}_{yq}\bar{\mathbf{M}}_{qq}^{-1}\bar{\mathbf{P}}_q + \bar{\mathbf{M}}_{yq}\bar{\mathbf{M}}_{qq}^{-1}\mathbf{\Phi}_{\bar{\mathbf{q}}}^T(\mathbf{\Phi}_{\bar{\mathbf{q}}}\bar{\mathbf{M}}_{qq}^{-1}\mathbf{\Phi}_{\bar{\mathbf{q}}}^T)^{-1}(\mathbf{\Phi}_{\bar{\mathbf{q}}}\bar{\mathbf{M}}_{qq}^{-1}\bar{\mathbf{P}}_q - \bar{\gamma}) \tag{14.7}$$

Since the virtual base body and the original chassis body are connected with a fixed joint, the equations of motion for the chassis body are obtained by simply adding the effective mass matrix in Eq. (14.6) and the effective force vector in Eq. (14.7) to the original chassis body equations of motion:

$$(\hat{\mathbf{M}}_0 + \breve{\mathbf{M}}^c)\dot{\hat{\mathbf{Y}}}_0 = (\hat{\mathbf{Q}}_0 + \breve{\mathbf{P}}^c) \tag{14.8}$$

where $\hat{\mathbf{M}}_0$ and $\hat{\mathbf{Q}}_0$ are the mass matric and generalized force vector of the chassis.

14.2.2 Full Robot Vehicle Model

The equations of motion for the full robot vehicle model can be easily constructed by synthesizing six of the 1/6 robot vehicle models using the subsystem synthesis method as shown in Fig. 14.3.

The chassis body equations of motion for the full robot vehicle model can now be obtained as shown in Eq. (14.9) by simply adding the effective mass matrices and the effective force vectors from all six suspension subsystems:

$$\left(\hat{\mathbf{M}}_0 + \sum_{i=1}^{6}\breve{\mathbf{M}}_i^c\right)\dot{\hat{\mathbf{Y}}}_0 = \left(\hat{\mathbf{Q}}_0 + \sum_{i=1}^{6}\breve{\mathbf{P}}_i^c\right) \tag{14.9}$$

As described in the previous section, the suspension subsystem equations of motion can be in exactly the same form as Eqs. (14.2) and (14.3):

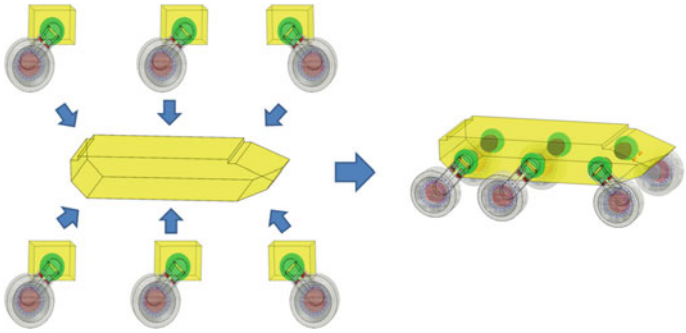


Fig. 14.3 A full robot vehicle model by synthesizing six 1/6 vehicle models

$$\boldsymbol{\lambda}_i = \left(\boldsymbol{\Phi}_{\bar{\mathbf{q}}_i}\bar{\mathbf{M}}_{q_iq_i}^{-1}\boldsymbol{\Phi}_{\bar{\mathbf{q}}_i}^T\right)^{-1}\left\{\boldsymbol{\Phi}_{\bar{\mathbf{q}}_i}\bar{\mathbf{M}}_{qq_i}^{-1}\left(\bar{\mathbf{P}}_{q_i} - \bar{\mathbf{M}}_{yq_i}^T\dot{\hat{\mathbf{Y}}}_0\right) - \bar{\boldsymbol{\gamma}}_i\right\} \quad \text{for } i = 1, 2, \ldots, 6 \tag{14.10}$$

$$\bar{\mathbf{M}}_{q_iq_i}\ddot{\bar{\mathbf{q}}}_i = \bar{\mathbf{P}}_{q_i} - \mathbf{M}_{yq_i}^T\dot{\hat{\mathbf{Y}}}_0 - \boldsymbol{\Phi}_{\bar{\mathbf{q}}_i}^T\boldsymbol{\lambda}_i \qquad \text{for } i = 1, 2, \ldots, 6 \tag{14.11}$$

In order to see the effectiveness of the subsystem synthesis method, the equations of motion for the full robot vehicle model using the conventional method [24] with joint coordinates can be derived:

$$\begin{bmatrix} \mathbf{M}_0 & \bar{\mathbf{M}}_{\mathbf{yq_1}} & \mathbf{0} & \cdots & \bar{\mathbf{M}}_{\mathbf{yq_6}} & \mathbf{0} \\ \bar{\mathbf{M}}_{\mathbf{yq_1}}^{\mathbf{T}} & \bar{\mathbf{M}}_{\mathbf{q_1q_1}} & \boldsymbol{\Phi}_{\mathbf{q_1}}^T & & & \\ \mathbf{0} & \boldsymbol{\Phi}_{\mathbf{q_1}} & \mathbf{0} & & & \\ \vdots & & & \ddots & & \\ \bar{\mathbf{M}}_{\mathbf{yq_6}}^{\mathbf{T}} & & & & \bar{\mathbf{M}}_{\mathbf{q_6q_6}} & \boldsymbol{\Phi}_{\mathbf{q_1}}^T \\ \mathbf{0} & & & & \boldsymbol{\Phi}_{\mathbf{q_6}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\hat{\mathbf{Y}}}_0 \\ \ddot{\mathbf{q}}_\mathbf{1} \\ \boldsymbol{\lambda}_\mathbf{1} \\ \vdots \\ \ddot{\mathbf{q}}_\mathbf{6} \\ \boldsymbol{\lambda}_\mathbf{6} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_{q_1} \\ \boldsymbol{\gamma}_\mathbf{1} \\ \vdots \\ \mathbf{P}_{q_6} \\ \boldsymbol{\gamma}_\mathbf{6} \end{bmatrix} \tag{14.12}$$

The dimensions of the inertia matrix of Eq. (14.12) are 36 by 36 for the full robot vehicle model. The dimensions of the $\bar{\mathbf{M}}_{qq}$ matrix in Eq. (14.11) are three by three, and the dimensions of the $(\boldsymbol{\Phi}_q\mathbf{M}_{qq}^{-1}\boldsymbol{\Phi}_q^T)$ in Eq. (14.10) are two by two. Thus, in the subsystem synthesis method, it is only necessary to solve linear equations with dimension 2 and then linear equations with dimension 3 to obtain the Lagrange multipliers and the joint accelerations, respectively, for each of the suspension subsystems. For the acceleration of the chassis body, linear equations with dimension 6 also have to be solved in the subsystem synthesis method. However, in the conventional method, 36 linear equations must be solved for the full robot vehicle model. Thus, the subsystem synthesis method is more computationally efficiency than the conventional method.

Another attractive feature of the subsystem synthesis method is the extendibility from the programming point of view. If one more subsystem is added to the chassis

body physically, then it is very easy to adopt this in the chassis equations of motion in Eq. (14.9) by adding the effective mass matrix and the effective force vector into the program. In contrast, it is difficult to extend within the program using the conventional method as shown in Eq. (14.12).

14.3 Real-Time Model with Suspension Model Simplification

Different suspension models can be easily adopted in the subsystem synthesis method on the condition that the effective mass matrices in Eq. (14.6) and the effective force vectors in Eq. (14.7) are properly generated. In order to improve the efficiency, the suspension model can be simplified using a single load arm (the housing) with a non-linear rotational spring as shown in Fig. 14.4. To obtain the characteristics of the non-linear rotational spring, a torque-angle curve was first obtained from the original suspension model described in Sect. 14.2, using the ADAMS program [22]. Then, a fifth order polynomial function has been introduced to approximate the torque-angle curve through the curve fitting technique as shown in Fig. 14.5.

If the conventional joint coordinate formulation [24] is used, the equations of motion for the suspension subsystems with the virtual base body can be derived as:

$$\begin{bmatrix} \bar{\bar{\mathbf{M}}}_{yy} & \bar{\bar{\mathbf{M}}}_{yq} \\ \bar{\bar{\mathbf{M}}}_{yq}^{T} & \bar{\bar{\mathbf{M}}}_{qq} \end{bmatrix} \begin{bmatrix} \dot{\hat{\mathbf{Y}}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \bar{\bar{\mathbf{P}}}_y \\ \bar{\bar{\mathbf{P}}}_q \end{bmatrix} \tag{14.13}$$

where $\dot{\hat{\mathbf{Y}}}_0$ is the chassis' state acceleration vector, $\ddot{\mathbf{q}}$ is the joint acceleration vector associated with the suspension subsystem, $\bar{\bar{\mathbf{M}}}_{yy}$, $\bar{\bar{\mathbf{M}}}_{yq}$ and $\bar{\bar{\mathbf{M}}}_{qq}$ are inertia matrices, and $\bar{\bar{\mathbf{P}}}_y$ and $\bar{\bar{\mathbf{P}}}_q$ are the generalized force vectors. In this simplified model, $\ddot{\mathbf{q}}$ is the revolute joint acceleration and is the scalar variable. From the second row of Eq. (14.13), the equation of motion for the subsystem is shown as Eq. (14.14);

$$\bar{\bar{\mathbf{M}}}_{qq}\ddot{\mathbf{q}} = \bar{\bar{\mathbf{P}}}_q - \bar{\bar{\mathbf{M}}}_{yq}^{T}\dot{\hat{\mathbf{Y}}}_0 \tag{14.14}$$

After obtaining the acceleration expression $\ddot{\mathbf{q}}$ in terms of $\dot{\hat{\mathbf{Y}}}_0$ from Eq. (14.14) and substituting this expression into the first row of Eq. (14.13), the reduced form of the virtual base body equations of motion is obtained as:

$$\breve{\mathbf{M}}^{c}\dot{\hat{\mathbf{Y}}}_0 = \breve{\mathbf{P}}^{c} \tag{14.15}$$

where,

$$\breve{\mathbf{M}}^{c} = \bar{\bar{\mathbf{M}}}_{yy} - \bar{\bar{\mathbf{M}}}_{yq}\bar{\bar{\mathbf{M}}}_{qq}^{-1}\bar{\bar{\mathbf{M}}}_{yq}^{T} \tag{14.16}$$

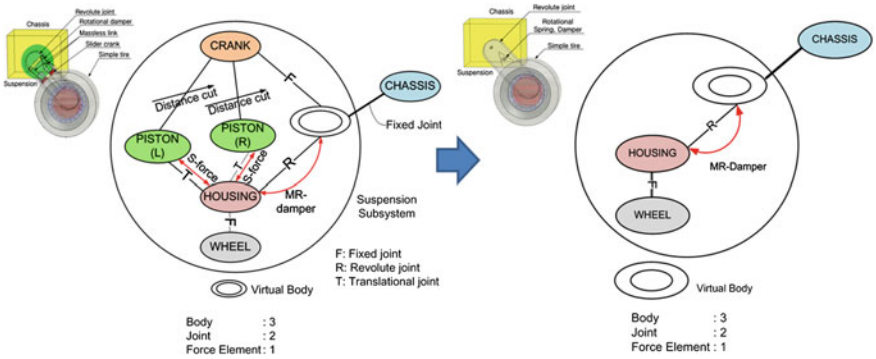


Fig. 14.4 Suspension model simplification

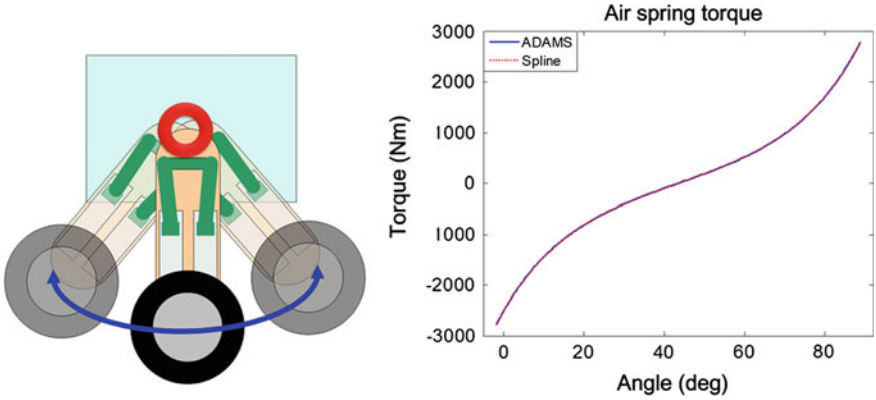


Fig. 14.5 Simplified suspension model with equivalent rotational stiffness

$$\breve{\bar{\mathbf{P}}}^c = \bar{\bar{\mathbf{P}}}_y - \bar{\bar{\mathbf{M}}}_{yq}\bar{\bar{\mathbf{M}}}_{qq}^{-1}\bar{\bar{\mathbf{M}}}_{yq}^T \tag{14.17}$$

The expressions of the effective mass $\breve{\bar{\mathbf{M}}}^c$ and the effective force vector $\breve{\bar{\mathbf{P}}}^c$ are much simpler than those of the original model in Eqs. (14.6) and (14.7). The chassis body equations of motion for the full robot vehicle model are in exactly the same form as the one in Eq. (14.9), even though the simplified suspension model is used:

$$(\hat{\mathbf{M}}_0 + \sum_{i-1}^{n} \breve{\bar{\mathbf{M}}}_i^c)\dot{\hat{\mathbf{Y}}}_0 = (\hat{\mathbf{Q}}_0 + \sum_{i-1}^{n} \breve{\bar{\mathbf{P}}}_i^c) \tag{14.18}$$

Once the acceleration of $\dot{\hat{\mathbf{Y}}}_0$ is obtained by solving Eq. (14.18), then the subsystem equations of motion can be solved using the following equations.

$$\bar{\bar{\mathbf{M}}}_{qq_i}\ddot{\mathbf{q}}_i = \bar{\bar{\mathbf{P}}}_{qi} - \bar{\bar{\mathbf{M}}}_{yqi}^{T}\dot{\bar{\mathbf{Y}}}_0 \qquad \text{for } i = 1, 2, \ldots, 6 \tag{14.19}$$

The dimensions of $\breve{\mathbf{M}}_{qqi}$ are only one by one. The matrix inversion procedure is not necessary. Eq. (14.18) has the same form as Eq. (14.9). The only differences are the expressions of the effective mass matrices and the effective force vectors. Thus, on the condition that the effective mass matrices and effective force vectors are correctly computed, the subsystem model can be easily replaced by other types of subsystem model. Using the simplified model, the computational amount can be drastically reduced as shown in Eqs. (14.16), (14.17) and (14.19). The efficiency with numerical simulation will be shown later in Sect. 14.5.

14.4 Subsystem Synthesis Method with Different Integrators

For a real-time model, it is not only important to have an efficient formulation for generating equations of motion but also to have a stable numerical integration method. For real-time integration, an inexpensive, accurate, stable, and fixed integration formula, which has the same amount of computational costs in each integration step, must be used [5]. Explicit multi-step methods are known to be accurate and inexpensive. However, they may not offer good stability. If the equations of motion are stiff, a very small stepsize must be used due to the narrow stability region of the explicit multi-step integrator [1]. In contrast, implicit integrators have much better numerical stability than explicit integrators. However, implicit integrators require an iterative solution procedure for solving non-linear equations. The iterative solution procedure with a convergence criterion fails to maintain the same computational cost for each time step, even though a fixed stepsize is used. Thus, it is also necessary to fix a number of iterations that is large enough to satisfy the convergence criterion.

In the subsystem synthesis method, different integration formulas can be applied to the subsystem equations of motion and to the chassis body equations of motion. Especially for the vehicle system, the suspension subsystem experiences high frequency excitation from rough terrain. However, the excitation to the chassis from the suspension system is alleviated due to the spring-damper in the suspension. In this section, a subsystem synthesis method with two different integrators is presented. The subsystem synthesis method with an explicit integrator is presented first in Sect. 14.4.1. The subsystem synthesis method with an explicit integrator for the chassis body equations of motion and an implicit integrator for each of the subsystem equations of motion is described in Sect. 14.4.2.

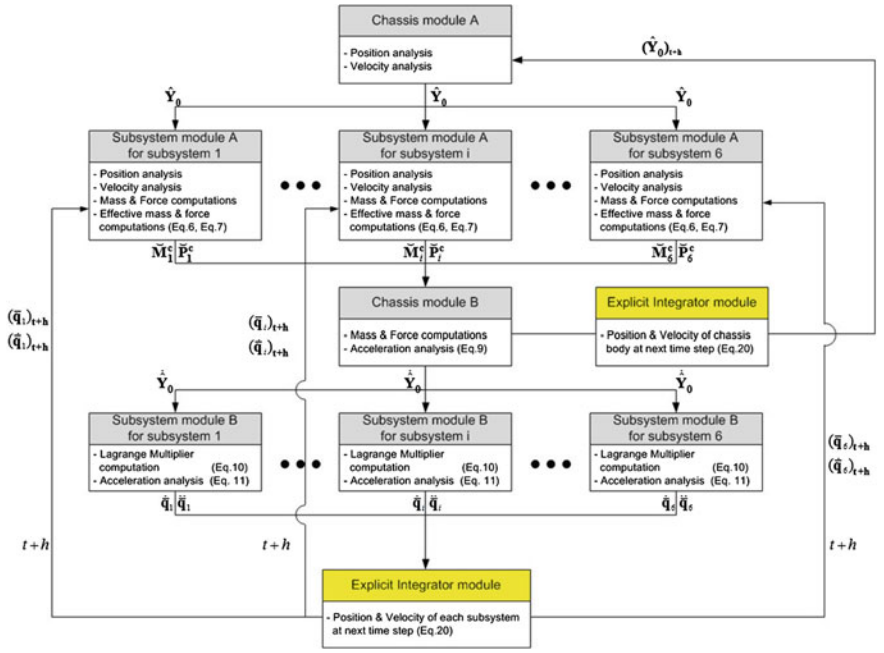


Fig. 14.6 Flow chart of the subsystem synthesis method with an explicit integrator

14.4.1 Explicit Integration Method

In order to apply the explicit integration formula for the subsystem synthesis method, the chassis' equations of motion can be solved for the acceleration of the chassis $\dot{\hat{\mathbf{Y}}}_{\mathbf{0}}$ from Eq. (14.9). Then, the explicit integration formula can be applied to predict the position and the velocity of the chassis for the next time step. Once the acceleration of chassis $\dot{\hat{\mathbf{Y}}}_{\mathbf{0}}$ is obtained, the Lagrange multiplier vectors and joint acceleration vectors in each suspension subsystem can be consequently computed from Eqs. (14.10) and (14.11). Acceleration of the joint coordinates can also now be integrated using an explicit integration formula for the position and the velocity of the next time step. In this chapter, the Adams-Bashforth 3rd order explicit formula [1] has been employed as:

$$\bar{\bar{\mathbf{y}}}_{\mathbf{t+1}} = \bar{\bar{\mathbf{y}}}_{\mathbf{t}} + \frac{\mathbf{h}}{\mathbf{12}}\left(23\dot{\bar{\bar{\mathbf{y}}}}_{\mathbf{t}} - 16\dot{\bar{\bar{\mathbf{y}}}}_{\mathbf{t-1}} + 5\dot{\bar{\bar{\mathbf{y}}}}_{\mathbf{t-2}}\right) \tag{14.20}$$

where $\bar{\bar{\mathbf{y}}}$ is the composite vector that contains the position and the velocity vector of the system.

Figure 14.6 shows the flow chart of the solution procedure from the subsystem synthesis method with an explicit integrator. In the Chassis Module A, the positions

and the velocity of the chassis body are computed. The composite vector $\hat{\mathbf{Y}}_\mathbf{0}$ of the chassis, which contains the position vector and the velocity state vector, is transferred from Chassis Module A to each of the modules of Subsystem Module A. In Subsystem Module A, the position vector $\mathbf{q}_i$ and the velocity vector $\dot{\mathbf{q}}_i$ of each subsystem are computed. The effective mass matrix $\breve{\mathbf{M}}_i^c$ in Eq. (14.6) and the effective force vector $\breve{\mathbf{P}}_i^c$ in Eq. (14.7) are also computed for each subsystem. The effective mass matrices and the effective force vectors from all subsystem modules are transmitted to Chassis Module B. In Chassis Module B, the chassis' equations of motion are formed and the acceleration of chassis $\dot{\hat{\mathbf{Y}}}_\mathbf{0}$ is evaluated by solving Eq. (14.9). $\dot{\hat{\mathbf{Y}}}_\mathbf{0}$ is then transferred into the each of the modules of Subsystem Module B. Also, the acceleration of chassis $\dot{\hat{\mathbf{Y}}}_\mathbf{0}$ is moved to the explicit integrator module to predict the position and velocity of the chassis for the next time step. In Subsystem Module B, the Lagrange multiplier associated with the subsystem is computed using Eq. (14.10), and the joint acceleration vector of the subsystem is also obtained by solving Eq. (14.11). Joint acceleration vectors from each of the modules of Subsystem Module B can be integrated for the next step using an explicit integrator. Due to the modular structure of the subsystem synthesis method, it is possible to apply different explicit integrators to the subsystem modules and the chassis module, as shown in this flow chart:

14.4.2 Explicit–Implicit Integration Method

For vehicle systems, a suspension system is needed to reduce the vertical wheel load variation and to isolate the road input. The suspension system can experience high frequency excitation from the road input. In contrast, the chassis frame can have a reduced frequency due to the suspension spring-dampers. If an explicit integration is applied especially to the off-road vehicle model, a very small stepsize must be used due to the high frequency excitation of the wheel and suspension subsystem. To alleviate this problem, the implicit integration formula, which has better stability than the explicit one, must be used. Due to the modular feature of the subsystem synthesis method, it is possible to apply an implicit integrator to the subsystem equations of motion that contain the stiff element or high frequency content and to impose an explicit integrator separately on the chassis' equations of motion [10, 12, 13].

In this chapter, the HHT integration formula [11] is employed as an implicit integrator. If the HHT-α integration method is applied to the subsystem equations of motion shown in Eq. (14.11), the complemented equations of motion, considering the numerical damping effect, are obtained, as shown in Eq. (14.21).

$$\boldsymbol{\Psi} \equiv (\bar{\mathbf{M}}_{\mathrm{yq}}^{\mathrm{T}} \dot{\hat{\mathbf{Y}}}_0)_{n+1} + (\bar{\mathbf{M}}_{\mathrm{qq}} \ddot{\mathbf{q}})_{n+1} + (1+\alpha)(\boldsymbol{\Phi}_{\mathrm{q}}^{\mathrm{T}} \boldsymbol{\lambda} - \bar{\mathbf{P}}_{\mathbf{q}})_{n+1} - \alpha(\boldsymbol{\Phi}_{\mathrm{q}}^{\mathrm{T}} \boldsymbol{\lambda} - \bar{\mathbf{P}}_{\mathbf{q}})_n = \mathbf{0} \tag{14.21}$$

In the HHT-α method, the Newmark formula is utilized for the joint velocity and the joint position for the next time step t_{n+1} as:

$$\dot{\bar{\mathbf{q}}}_{n+1} = \dot{\bar{\mathbf{q}}}_n + (1-\gamma)h\ddot{\bar{\mathbf{q}}}_n + \gamma h\ddot{\bar{\mathbf{q}}}_{n+1} \tag{14.22}$$

$$\bar{\mathbf{q}}_{n+1} = \bar{\mathbf{q}}_n + h\dot{\bar{\mathbf{q}}}_n + \frac{h^2}{2}(1-2\beta)\ddot{\bar{\mathbf{q}}}_n + \beta h^2\ddot{\bar{\mathbf{q}}}_{n+1} \tag{14.23}$$

where, h is a stepsize and the subscripts denote the discrete time steps. In the HHT-α method, the desirable level of numerical damping in the system can be adjusted with the integration parameter α.

$$\alpha \in \left[-\tfrac{1}{3}\ 0\right], \beta = \frac{(1-\alpha)^2}{4}, \gamma = \frac{1-2\alpha}{2} \tag{14.24}$$

An iterative method, such as the Newton-Raphson algorithm, can be applied to solve the resulting systems of nonlinear equations [23] shown in Eq. (14.21) as:

$$\begin{bmatrix} \frac{\boldsymbol{\Psi}_{\ddot{\bar{\mathbf{q}}}}}{1+\alpha} & \boldsymbol{\Phi}_{\bar{q}}^{\mathbf{T}} \\ \boldsymbol{\Phi}_{\bar{\mathbf{q}}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\bar{\mathbf{q}}}^{(k)} \\ \Delta\bar{\boldsymbol{\lambda}}^{(k)} \end{bmatrix} = \begin{bmatrix} -\frac{\boldsymbol{\Psi}}{1+\alpha} \\ -\frac{\boldsymbol{\Phi}}{\beta h^2} \end{bmatrix} \tag{14.25}$$

$$\text{with, } \begin{matrix} \ddot{\bar{\mathbf{q}}}^{(k+1)} = \ddot{\bar{\mathbf{q}}}^{(k)} + \Delta\ddot{\bar{\mathbf{q}}}^{(k)} \\ \bar{\boldsymbol{\lambda}}^{(k+1)} = \bar{\boldsymbol{\lambda}}^{(k)} + \Delta\bar{\boldsymbol{\lambda}}^{(k)} \end{matrix}$$

where, (k) represents the iteration counts for the Newton-Raphson method. In Eq. (14.25), $\boldsymbol{\Psi}_{\ddot{\bar{\mathbf{q}}}}$ is the system Jacobian matrix that can be expressed as the following equation:

$$\boldsymbol{\Psi}_{\ddot{\bar{\mathbf{q}}}} = \bar{\mathbf{M}}_{qq} + \beta h^2(\boldsymbol{\Psi})_{\bar{\mathbf{q}}} + \gamma h(\boldsymbol{\Psi})_{\dot{\bar{\mathbf{q}}}} + (\boldsymbol{\Psi})_{\ddot{\bar{\mathbf{q}}}} \tag{14.26}$$

The computation of the system Jacobian matrix is very complicated. To derive the system Jacobian expression, MAPLE [20] is used in this chapter.

For the simplified suspension model described in Sect. 14.3, if the HHT–α method is applied, the complemented equations of motion for the subsystem equations of motion are as follows:

$$\bar{\boldsymbol{\Psi}} \equiv (\bar{\bar{\mathbf{M}}}_{yq}^{T}\dot{\hat{\mathbf{Y}}}_0)_{n+1} + (\bar{\bar{\mathbf{M}}}_{qq}\ddot{\bar{\mathbf{q}}})_{n+1} - (1+\alpha)(\bar{\bar{\mathbf{P}}}_q)_{n+1} + \alpha(\bar{\bar{\mathbf{P}}}_q)_n = \mathbf{0} \tag{14.27}$$

An iterative method, such as the Newton-Raphson algorithm, can also be applied to solve the resulting systems of nonlinear equations for Eq. (14.27):

$$\begin{aligned} \bar{\boldsymbol{\Psi}}_{\ddot{\bar{\mathbf{q}}}}\Delta\ddot{\bar{\mathbf{q}}}^{(k)} &= -\bar{\boldsymbol{\Psi}} \\ \ddot{\bar{\mathbf{q}}}^{(k+1)} &= \ddot{\bar{\mathbf{q}}}^{(k)} + \Delta\ddot{\bar{\mathbf{q}}}^{(k)} \end{aligned} \tag{14.28}$$

where (k) represents the iteration counts for the Newton-Raphson method. In Eq. (14.28), $\bar{\boldsymbol{\Psi}}_{\ddot{\mathbf{q}}}$ is the system Jacobian matrix that can be expressed as the following equation:

$$\bar{\boldsymbol{\Psi}}_{\ddot{\mathbf{q}}} = \bar{\mathbf{M}}_{qq} + \beta h^2 (\boldsymbol{\Psi})_{\bar{\mathbf{q}}} + \gamma h (\boldsymbol{\Psi})_{\dot{\bar{\mathbf{q}}}} + (\boldsymbol{\Psi})_{\ddot{\bar{\mathbf{q}}}} \tag{14.29}$$

To derive the expression of the system Jacobian matrix effectively, the symbolic code generator MAPLE is also used [20].

Figure 14.7 shows the flow chart of the solution procedure from the subsystem synthesis method with an explicit–implicit integrator. In the initial computation box, acceleration analysis of the full robot vehicle is carried out using Eqs. (14.9) and (14.11), since the acceleration values are required in Eq. (14.21). The composite vectors $\hat{\mathbf{Y}}_{\mathbf{0}}$ and $\dot{\hat{\mathbf{Y}}}_{\mathbf{0}}$ and joint position, velocity and acceleration are transferred to each of the subsystem modules. In the Subsystem Module, the joint positions and joint velocities for the next time step are estimated using Taylor series expansion. These estimated values are used for computing mass matrices and force vectors in Eq. (14.21) in the first iteration of the Newton Raphson. The complemented equation $\boldsymbol{\Psi}$ shown in Eq. (14.21) is then evaluated. System Jacobian in Eq. (14.26) is also computed. Afterward, the system equation in Eq. (14.25) is solved to obtain the joint acceleration. During iterative procedure, joint position and joint velocity are also updated using the Newmark formulas in Eqs. (14.22) and (14.23). Once the convergence criterion $\|\boldsymbol{\Psi}\|$ is satisfied, subsystem analysis is completed for joint positions, joint velocities and joint accelerations. With these converged variables, the effective mass matrices in Eq. (14.6) and the effective force vectors in Eq. (14.7) are computed. The effective mass matrices and the effective force vectors from each subsystem are then transmitted to Chassis Module B. In Chassis Module B, the chassis' equations of motion are formed, and the acceleration of the chassis $\dot{\hat{\mathbf{Y}}}_{\mathbf{0}}$ is evaluated by solving Eq. (14.9). The acceleration of the chassis $\dot{\hat{\mathbf{Y}}}_{\mathbf{0}}$ is moved to the explicit integrator module to predict the position and velocity of the chassis for the next time step. It is noted that although the implicit formula for the subsystem equations of motion requires the acceleration of the chassis at the current step, the acceleration from the previous step is used instead by assuming that the acceleration variation is small.

14.5 Real-Time Simulations of the Unmanned Robot Vehicle with Rough Terrain

The full robot vehicle model was created and implemented using the C language. Once the 1/6 robot vehicle model was implemented, then the full robot vehicle model can be constructed very effectively, since the subsystem synthesis method provides effective means to expand from the 1/6 robot vehicle model to the full robot vehicle model as described in Sect. 14.2. Table 14.1 shows the inertia properties of the full

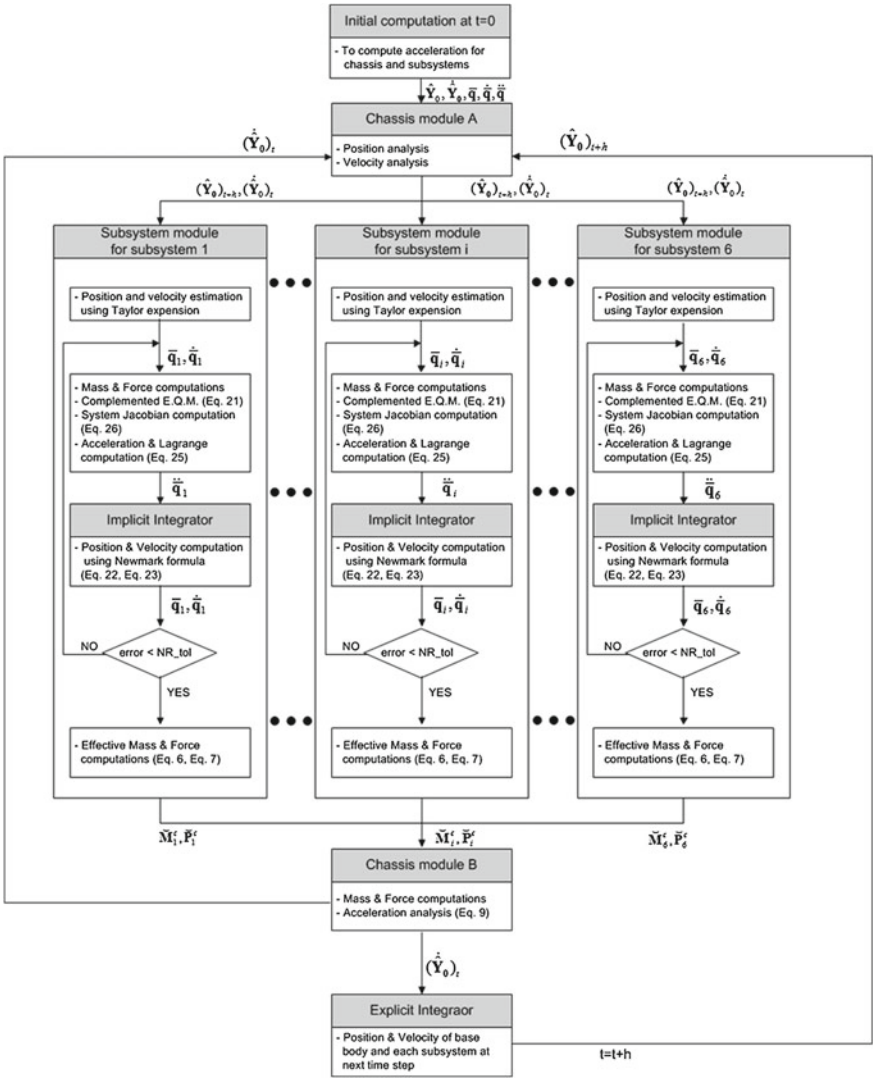


Fig. 14.7 Flow chart of the subsystem synthesis method with an explicit–implicit integrator

robot vehicle model. Table 14.2 also shows the parameters associated with the force elements in the suspension subsystem shown in Fig. 14.2.

In order to investigate the efficiency of the subsystem synthesis method relative to the conventional method, a full robot vehicle model was also created using the conventional method shown in Eq. (14.12) and implemented using the C language. Rough terrain simulations have been carried out with the full robot vehicle models from the two different formulations. The vehicle speed was 10 km/h, and PSD level

Table 14.1 Inertia properties of the robot vehicle

Inertia properties	Mass (kg)	Inertia (kg m^2)		
		$\mathbf{I}_{x'x'}$	$\mathbf{I}_{y'y'}$	$\mathbf{I}_{z'z'}$
Chassis	440	512.736	460.446	861.830
Housing and wheel	16.78	0.536	0.493	0.067

Table 14.2 Force element properties of suspension subsystem

Element properties	Values
Air spring stiffness	$2 \times \left(\frac{\pi \times 60^2}{4}\right) \times \left(\frac{0.07589}{l}\right)^{1.8}$ Nm/deg
MR damper damping coefficient	10.47 Nm s/deg
Tire vertical stiffness	200,000 N/m

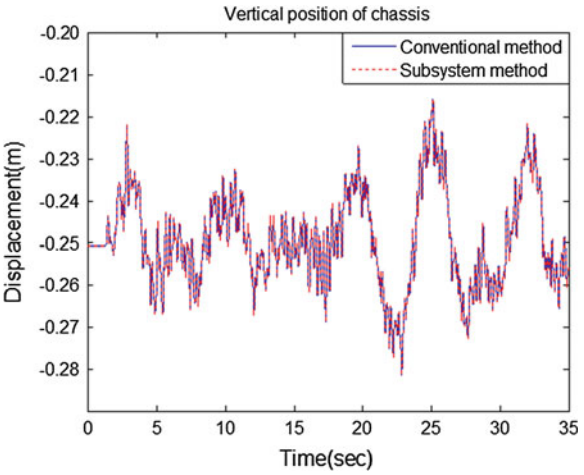


Fig. 14.8 Vertical positions of the chassis from models with subsystem synthesis method and with the conventional method

D road roughness was chosen [7]. Figure 14.8 shows the vertical positions of the chassis with the two different models. Essentially identical results were obtained from the two different models.

Figure 14.9 shows the CPU time comparison between the subsystem synthesis method and the conventional method. In this simulation, the Adams-Bashforth 3rd order explicit integrator was used with a fixed stepsize of 0.8 ms. As a computational platform, a PC with an Intel CoreTM i5-3,570 K 3.40 GHz CPU, and 3,392 MB RAM was utilized for a 35-s simulation time. The model with the conventional method took 9.867 s whereas the model with the subsystem synthesis method took only 4.439 s. Thus, the real-time simulation was achieved with both models. However, the

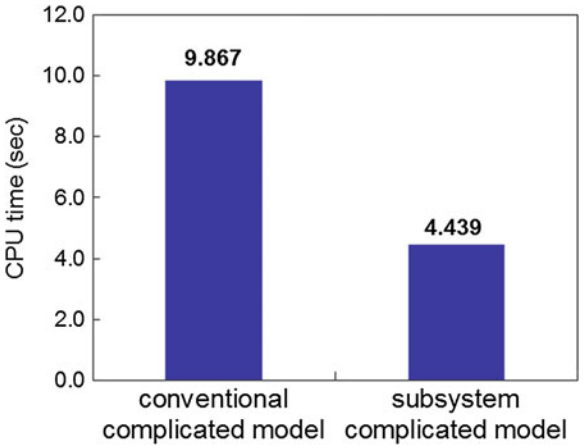


Fig. 14.9 CPU time comparison between the subsystem synthesis method and the conventional method

subsystem synthesis method was about 2.2 times faster than the conventional method because it solved several small equations of motion rather than a large matrix form of equations of motion.

In order to examine the efficiency of to the suspension model simplification described in Sect. 14.3, a full robot vehicle model with a simplified suspension subsystem was also developed. To develop this model, only the subsystem model was implemented at first, and then the original subsystem modules were replaced by the simplified suspension subsystem modules without altering the program structure. The simulations were carried out with the same conditions as the previous ones. Figure 14.10 shows the roll angles of the chassis from the model with the original suspension (the complicated model) and from the model with the simplified suspension. Essentially identical responses were obtained. Thus, the model with the simplified suspensions has been validated.

Figure 14.11 shows the CPU processing time comparison from the two different suspension subsystem models. The model with the simplified suspension is about 1.9 times faster than the one with the original suspension. It is noted that the original suspension model contains multiple closed loops whereas the simplified model has an open loop system. Thus, computational saving are accomplished, as described in Sect. 14.3.

In order to investigate the effciency of the different integration methods, a full robot vehicle model was developed and implemented using the C language based on the explicit–implicit formulation described in Sect. 14.4.2. The full robot vehicle model with the simplified suspension was utilized in this comparison study. The simulation conditions for the rough terrain run were the same as previous ones. When the explicit integration method is applied to the model, the maximum stepsize is 0.8 ms. If a larger stepsize than 0.8 ms is used, unstable solutions are obtained. For

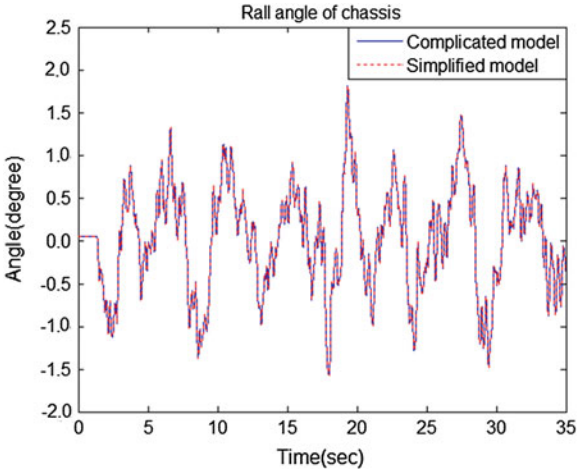


Fig. 14.10 Roll angles of the chassis from complicated and simplified subsystem models

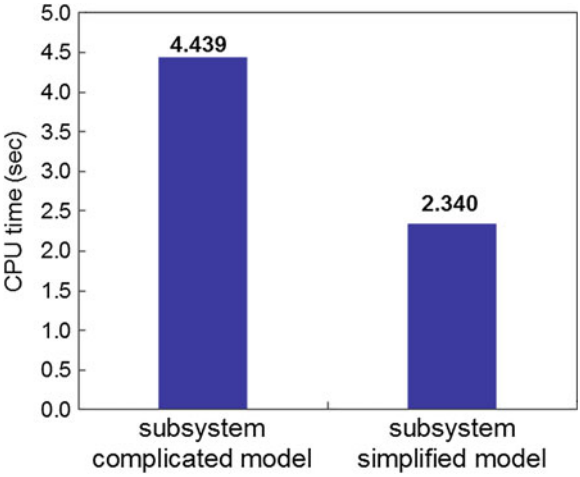


Fig. 14.11 CPU time comparison between models with complicated suspensions and with simplified suspension

the full robot vehicle model with the simplified suspsensions, the CPU time taken is only 2.340 s out of the 35-s simulation time.

Figure 14.12 shows the CPU time results according to the various stepsizes of the explicit–implicit integrator. If the explicit–implicit integrator is employed, a larger stepsize than 0.8 ms can be used to produce stable results. However, when the larger stepsize was used, the RMS errors also increased as shown in Fig. 14.13. In the RMS error computation, the acceleration responses from the model with the explicit

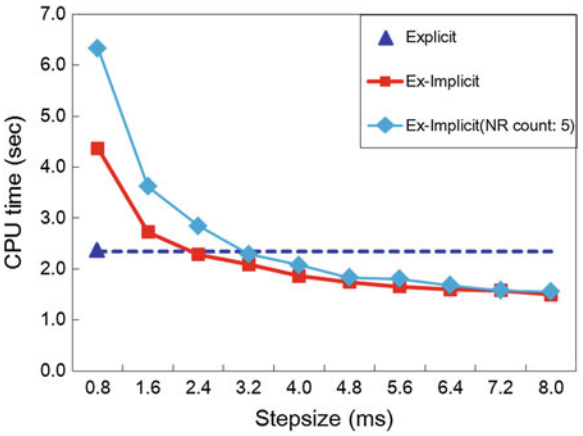


Fig. 14.12 CPU times with the various stepsizes of the explicit–implicit integrator

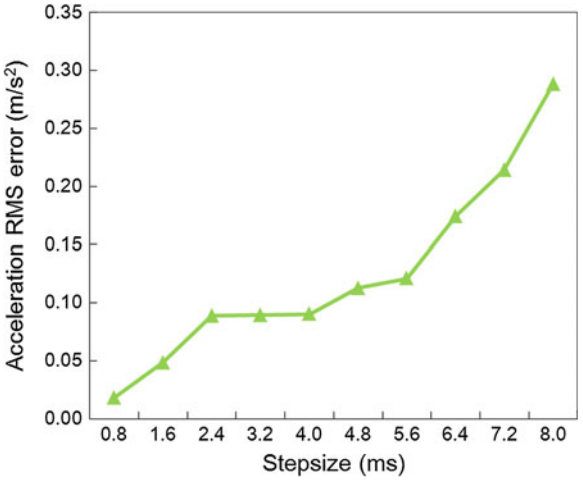


Fig. 14.13 RMS errors according to the various stepsizes

integration were treated as the reference solutions. When the stepsize was larger than 5.6 ms, the RMS errors drastically increased. Thus, a stepsize of 4.8 ms was chosen as the maximum stepsize that produces stable solutions without much loss of accuracy for the explicit–implicit integrator. In Table 14.3, the RMS errors for position, velocity, and acceration variables are also presented for the integration stepsize of 4.8 ms. In the subsystem synthesis method, since the subsystem equations of motion are solved independently, each subsystem has different convergence characteristics. Thus, different average Newton Raphson iteration counts are presented as shown in Table 14.3 for the subsystems (LF, LM, LR, RF, RM, and RR). For real-time

Table 14.3 Detailed CPU time results

Full car model Rough terrain Simulation: 35 s		Method		
		Explicit (reference)	Ex-implicit	Ex-implicit (NR count: 5)
Max. stepsize (ms)		0.8	4.8	4.8
RMS error	Pos. (m)	–	4.3833E−05	4.8259E−05
	Vel. (m/s)	–	0.0021	0.0021
	Acc. (m/s^2)	–	0.1128	0.1482
Ave. NR count		–	RF: 4.57	RF: 5
			RM: 2.98	RF: 5
			RR: 3.51	RR: 5
			LF: 3.61	LF: 5
			LM: 3.05	LM: 5
			LR: 4.24	LR: 5
CPU time (s)		2.340	1.746	1.833
Ratio of CPU time to simulation time (%)		6.69	4.99	5.24
Ratio among three methods		1	0.75	0.65

LF Left front suspension, *LM* Left middle suspension, *LR* Left rear suspension
RF Right front suspension, *RM* Right middle suspension, *RR* Right rear suspension

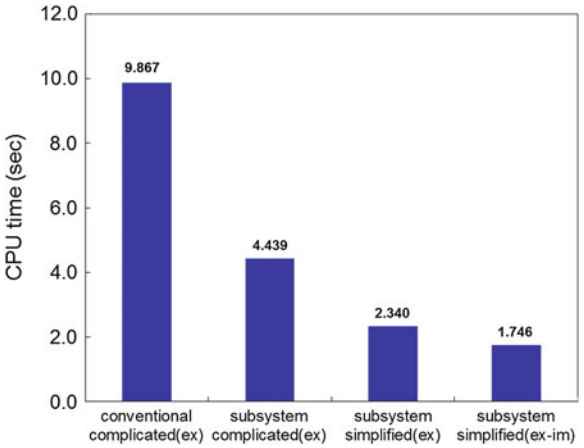


Fig. 14.14 CPU time comparison among different models based on formulations, model simplification, and integration methods

simulations, a fixed computational cost for each integration time step is required in the case of HILS. Simulations have been also carried out with a fixed Newton Raphson (NR) count of 5 in order to ensure the same computational cost for each time step. The CPU time results with various integration stepsizes for the model with a fixed NR count are also shown in Fig. 14.12. The model with a fixed NR count consumed

lager CPU times than the model without a fixed NR count in the smaller integration stepsizes. With smaller integration stepsizes, the solutions converged with less than five iterations. The larger integration stepsizes are used, the more iterations are necessary for convergence. Thus, when the stepsize of 4.8 ms was employed, the similar CPU times were required for the both models with and without a fixed NR count as shown in Table 14.3.

As a summary of the performances of the different models, Fig. 14.14 shows the CPU time comparison among the different models discussed in this chapter; i.e., the models with different formulations, the models with and without suspension model simplification, and the models with different integration methods. In the 35-s simulation, the model using the subsystem synthesis method improved efficiency by reducing the CPU time from 9.866 to 4.439 s. Using the simplified suspension model reduced the CPU time from 4.439 to 2.340 s additionally. Finally, if the explicit–implicit integration method is employed with an integration stepsize of 4.8 ms, we were able to reduce further the CPU time from 2.340 s to 1.746 s. Thus, if we apply the subsystem synthesis method to the full robot vehicle model with the simplified suspension model and also employ the explicit–implicit integrator, then we can achieve the real-time simulations of in which the CPU time requires only 5.24 % of the actual time.

14.6 Conclusions

Real-time multibody dynamics models for an unmanned robot vehicle have been developed. The unmanned robot vehicle consists of six identical suspension subsystems. The subsystem synthesis method, which is ideally suited for a vehicle system with several identical suspension subsystems, has been applied for real-time simulations. In order to improve the efficiency, the suspension subsystem model has been simplified. The original suspension model consists of an MR-rotary damper, air springs with a double slider-crank mechanism. The simplified suspension model has been created by replacing the air springs with the double slider-crank mechanism to a non-linear rotational spring. By taking advantage of the subsystem synthesis method, the full robot vehicle model has been effectively created by replacing the original suspension subsystem model to the simplified subsystem model. For the stable solutions with a large integration stepsize, the explicit–implicit integrator with the subsystem synthesis method has been also employed. Using rough terrain simulations, the efficiency of the created model has been investigated. For real-time simulations which require the same computational cost, a procedure for a fixed number of iterations in the implicit integration has been demonstrated. The CPU time results show that the model with the simplified suspension provides the best performance with the explicit–implicit integrator.

References

1. Atkinson KE (1978) An introduction to numerical analysis. Wiley, London
2. Bae DS, Haug EJ (1987) A recursive formulation for constrained mechanical system dynamics—part I :open-loop systems. Mech Struct Mach 15(3):359–382
3. Bae DS, Haug EJ (1987) A recursive formulation for constrained mechanical system dynamics—part II: closed-loop systems. Mechanics of Structures and Machines 15(4):481–506
4. Bae DS, Hwang RS, Haug EJ (1988) A recursive formulation for real-time dynamic formulation. In: Rao SS (ed) Advances in design automation (ASME), pp 499–508
5. Bayo E, Garcia de Jalon J, Avello A, Cuadrado J (1991) An efficient computational method for real time multibody dynamics simulation in fully Cartesian coordinates. Comput Method Appl Mech Eng 92:377–395
6. Bouhraoua A, Merah N, AlDajani M, ElShafei M (2010) Design and implementation of an unmanned ground vehicle for security applications. In: Proceedings of the 7th international symposium on mechatronics and its applications, Dhahran, Saudi Arabia
7. Choi GJ, Heo SJ (2006) Classification of the Korean road roughness. Trans KSAE 14(5): 115–120
8. Featherstone R (1983) The calculation of robot dynamics using articulated body inertias. Int J Rob Res 2:13–30
9. Featherstone R (1987) Robot dynamics algorithm. Kluwer Academic Publication, New York
10. Han JB, Jo JY, Kim S-S, Wang JH, Kim JY (2011) An explicit-implicit integration method for multibody dynamics model based on a subsystem synthesis method. In: Proceedings of multibody dynamics 2011 ECCOMAS thematic conference, Brussels, Belgium
11. Hibert HM, Hughes TJR, Talyor RL (1977) Improved numerical dissipation for time integration algorithms in structural dynamics. Earthq Eng Struct Dyn 5:283–292
12. Jo JY, Kim MH, Kim S-S (2012) Joint coordinate subsystem synthesis method with implicit integrator in the application to the unmanned military robot. EUROMECH Colloquium 524 multibody system modeling, control and simulation for engineering design
13. Jo JY, Kim S-S, Kim MH (2012) An explicit-implicit integration method for unmanned robot vehicle using the subsystem synthesis method based on joint coordinates. In: Proceedings of the 2nd joint international conference on multibody system dynamics, Stuttgart, Germany
14. Joo SH, Lee JH, Park YW, Yoo WS, Lee JH (2013) Real time traversability analysis to enhance rough terrain navigation for an 6×6 autonomous vehicle. J Mech Sci Technol 27(4):1125–1134
15. Kim S-S (2002) A subsystem synthesis method for efficient vehicle multibody dynamics. Multibody Syst Dyn 7(2):189–207
16. Kim S-S, Jeong WH, Jo JY, Wang JH (2011) Mutlibody vehicle dynamics analysis using an explicit-implict integrator with subsystem synthesis method. In: Proceedings on ASME international design engineering technical conferences and computers and information in engineering conference, Washington DC, USA
17. Kim S-S, Jeong WH, Jung DH, Choi HJ (2010) HIL-simulation for evaluation of intelligent chassis controller using real-time multibody vehicle dynamics model. In: Proceedings of the 1st joint international conference on multibody system dynamics, Lappeenranta, Finland
18. Kim S-S, Jeong WH, Kim MH, Han JB (2012) Comparative study on multibody vehicle dynamics models based on subsystem synthesis method using Cartesian and joint coordinates. Theoret Appl Mech Lett 2:063010
19. Kim S-S, Kim SW, Kang HC, Oh MC (2013) A remote operating system of an unmanned military robot for indoor test environment. In: Proceeding of ISR 2013
20. Maplesoft (2011) Maple TM 15 user manual. A division of Waterloo Maple Inc
21. Mogebbi A, Safaee S, Keshmiri M, Mohebbi S (2010) Design, simulation and manufacturing of a tracked surveillance unmanned ground vehicle. In: Proceedings of the IEEE international conference on robotics and biomimetices, Motreal, Canada
22. MSC Software (2013) Complete multibody dynamics analysis with adams. MSC Software Inc

23. Negrut D, Rampalli R, Ottarssom G, Sajdak A (2005) On the use of the HHT method in the context of index 3 differential algebraic equations of multibody dynamics. In: Proceedings of ASME IDETC/CIE, pp 1–12
24. Tsai FF, Huag EJ (1991) Real-time multibody system dynamic simulation, part I: a modified recursive formulation and topological analysis. Mech Struct Mach 19(1):99–127
25. Walker MW, Orin DE (1982) Efficient dynamic computer simulation of robotic mechanisms. J Dyn Syst Meas Control ASME 104:205–211

Chapter 15
History of Benchmark Problems in Multibody Dynamics

Werner Schiehlen

Abstract In computational mechanics, and in particular in multibody dynamics, benchmarks are very helpful to identify the strongs and weaks of algorithms, formalisms, and computer program packages with respect to applications in science and engineering. The history of benchmark problems is reviewed, some example including gyrodynamics, mechanisms, road and rail vehicles as well as flexible beams are presented. Future developments will be discussed.

15.1 Introduction

According to Wikipedia [1, 2] a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative *performance* of an object, normally by running a number of standard tests and trials against it. The term 'benchmark' is also utilized for the purposes of elaborately-designed benchmarking programs themselves. Benchmarking is often associated with assessing performance characteristics of computer hardware, for example, the floating point operation performance of a CPU. But the technique is also applicable to software. Software benchmarks are, for example, run against compilers or database management systems. Benchmarks provide a method of comparing the performance of various subsystems across different computer architectures. On the other hand test suites are a type of system intended to assess the *correctness* of software.

In software development, a test suite or *validation suite*, respectively, is a collection of test cases or *benchmark problems* that are intended to be used to test a software program to show that it has some specified set of behaviour. A test suite

W. Schiehlen (✉)
Institute of Engineering and Computational Mechanics, University of Stuttgart,
Pfaffenwaldring 9, 70550 Stuttgart, Germany
e-mail: werner.schiehlen@itm.uni-stuttgart.de

Z. Terze (ed.), *Multibody Dynamics*, Computational Methods in Applied Sciences 35,
DOI: 10.1007/978-3-319-07260-9_15,

often contains detailed instructions or goals for each collection of test cases and information on the system configuration to be used during testing. A group of test cases may also contain prerequisite states or steps, and descriptions of the following tests.

In multibody dynamics complete validation suites are not available but benchmark problems have been defined and successfully used. In this historical review a number of benchmark problems is presented from various engineering applications. Ideally, the benchmark problems are solved on the same computer by competitive multibody simulation software. But this is often not possible due to the availability of hardware and software resources. Nevertheless, the accuracy of the results and the efficiency of the computations provide an important information on performance and correctness for program developers and software users.

15.2 Classes of Benchmarks

There are four classes of benchmarks considered including gyrodynamics, mechanisms, vehicle dynamics and flexible multibody systems.

15.2.1 Gyrodynamics

The first class of benchmarks is related to the fundamentals of multibody systems originating from gyro dynamics. A gyroscope in Cardanic suspension represents a three-body system with springs as treated 1942 by Magnus [3], and it is shown in Fig. 15.1. Magnus presented the completely exact equations of motion shown in Fig. 15.2 in a compact form. The symbols in the equations of motion are defined as follows: momentum vector **J**, relative angular velocity vector **u'**, torque vector of the springs **M**, angles β and ϕ related to the inner ring and the rotor, respectively, indices 1, 2, 3 for the rotor, inner ring and outer ring. The indices A, B, C identify the axes of the outer ring, inner ring and rotor, some of them also known as Prandtl rotation axes. Furthermore, (*) means time derivative in the body fixed frames.

Later in 1966 Magnus [4, 5] discussed the stability behavior of a force-free asymmetric gyroscope represented by the Prandtl wheel, Fig. 15.3. The inertia distribution can be changed by fixing additional weights at the rotor bearings of the inner gimbal resulting in an increase θ of the corresponding moments of inertia. Obviously, the stability behavior of the three-body Prandtl wheel is deviating from a one-body Eulerian gyro featuring instability only about the axis of the middle moment of inertia. In particular, the domain $\theta_1 < \theta < \theta_2$ shows only one stable rotation while in the domain $\theta_3 < \theta < \theta_4$ there isn't any instability. These results summarized in Table 15.1 have been experimentally confirmed, and they can been used as reliable benchmarks in computational multibody dynamics.

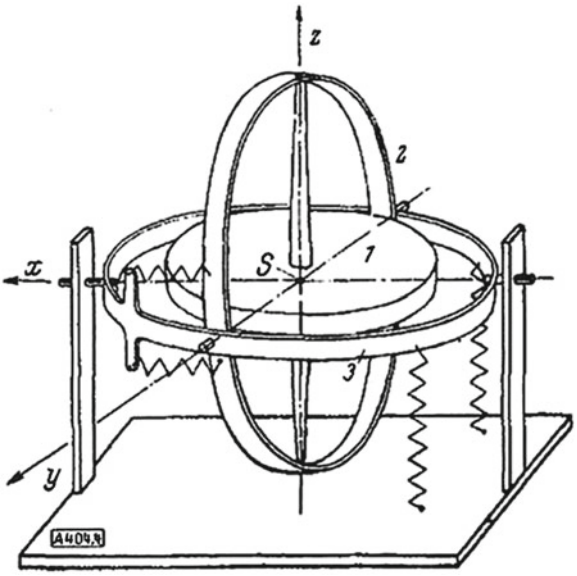


Fig. 15.1 Gyro as three-body system [3]

$$
\begin{aligned}
&(\overset{*}{\mathfrak{J}}_1)_A + (\mathfrak{J}_1)_C u'_B - (\mathfrak{J}_1)_B u'_C = \{M_{A_3} - (\overset{*}{\mathfrak{J}}_3)_{A_3} - [+(\overset{*}{\mathfrak{J}}_2)_{C_2} + (\mathfrak{J}_2)_{B_2} u'_{A_2} - (\mathfrak{J}_2)_{A_2} u'_{B_2}] \sin^2\beta\} \frac{\cos\varphi}{\cos\beta} \\
&\quad + M_{B_{23}} \sin\varphi - (\overset{*}{\mathfrak{J}}_2)_A - (\mathfrak{J}_2)_{C_2} (u'_2)_B + (\mathfrak{J}_2)_B u'_{C_2} \\
&(\overset{*}{\mathfrak{J}}_1)_B + (\mathfrak{J}_1)_A u'_C - (\mathfrak{J}_1)_C u'_A = -\{M_{A_3} - (\overset{*}{\mathfrak{J}}_3)_{A_3} - [(\overset{*}{\mathfrak{J}}_2)_{C_2} + (\mathfrak{J}_2)_{B_2} u'_{A_2} - (\mathfrak{J}_2)_{A_2} u'_{B_2}] \sin^2\beta\} \frac{\sin\varphi}{\cos\beta} \\
&\quad + M_{B_{23}} \cos\varphi - (\overset{*}{\mathfrak{J}}_2)_B - (\mathfrak{J}_2)_{A_2} (u'_2)_C + (\mathfrak{J}_2)_C u'_{A_2} \\
&(\overset{*}{\mathfrak{J}}_1)_C + (\mathfrak{J}_1)_B u'_A - (\mathfrak{J}_1)_A u'_B = 0
\end{aligned}
$$

Fig. 15.2 Equations of motion [3]

15.2.2 Mechanisms

A second class of benchmarks includes Andrew's seven-body mechanism also known as Andrew's squeezer mechanism proposed for early impact printers, Fig. 15.4. This benchmark was promoted by Giles [6] and Manning [7] in the late 1970s.

The mechanism performs a plan motion. The disassembled system has $7 \times 3 = 21$ degrees of freedom. The system is assembled by 10 bearings resulting in 20 constraints with 1 degree of freedom left. The mechanism was used 1990 as benchmark in the Multibody Systems Handbook [8] with a detailed description of all the kinematical and dynamical parameters. The angle β was used as generalized coordinate and the scalar symbolic equation of motion has been generated with the software NEWEUL where the other angles, e.g. $\gamma(\beta)$, serve as auxiliary variable. The time integration of the scalar ordinary differential equation (ODE) of the angle β and the

Fig. 15.3 Prandtl wheel [5]

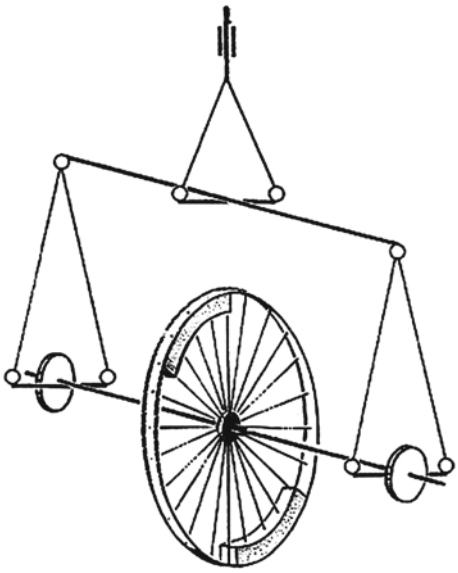


Table 15.1 Stability domains of the Prandtl wheel: stable (+) and unstable (−)

Prandtl rotation	$\theta < \theta_1$	$\theta_1 < \theta < \theta_2$	$\theta_2 < \theta < \theta_3$	$\theta_3 < \theta < \theta_4$	$\theta_4 < \theta$
No. 1	+	−	−	+	+
No. 2	+	+	+	+	−
No. 3	−	−	+	+	+

post-processing for $\gamma(\beta)$ result for a constant drive torque in the motion shown in Fig. 15.5.

Later on in 1996 Andrew's seven-body mechanism was widely applied for testing differential algebraic equation (DAE) solvers by Hairer and Wanner [9]. The mechanism has three closed kinematical loops connecting bearings O, A and B and resulting in six algebraic closing conditions. Simulation results are shown in Fig. 15.6, in particular the angle γ shows the same motion as depicted in Fig. 15.5.

Moreover, Hairer and Wanner [9] compare index 3, 2, and 1 formulations by work-precision diagrams plotting the computing time over the error of the components at time t = 0.03 s. It turns out that index 1 formulation with velocity stabilization provides the best results with respect to the accuracy achieved and the computing time required.

15.2.3 Vehicle Dynamics

A third class of benchmarks is devoted to vehicle dynamics. Kortuem and Sharp [10] published 1993 a book comparing vehicle dynamic software, and used the Iltis vehicle

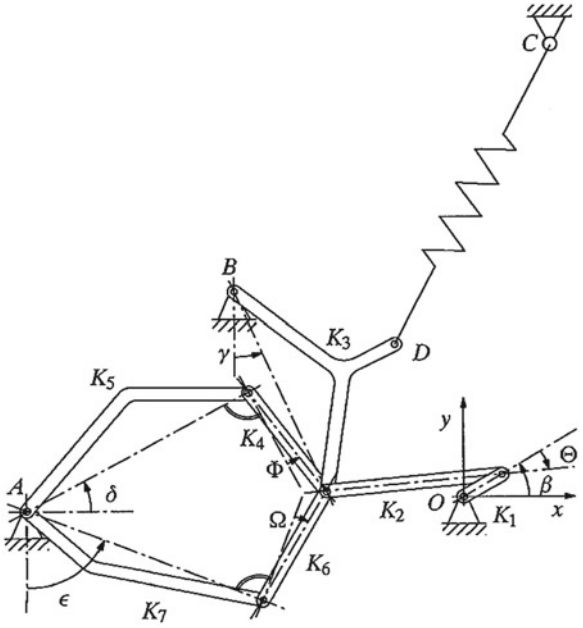


Fig. 15.4 Mechanism as seven-body system [8]

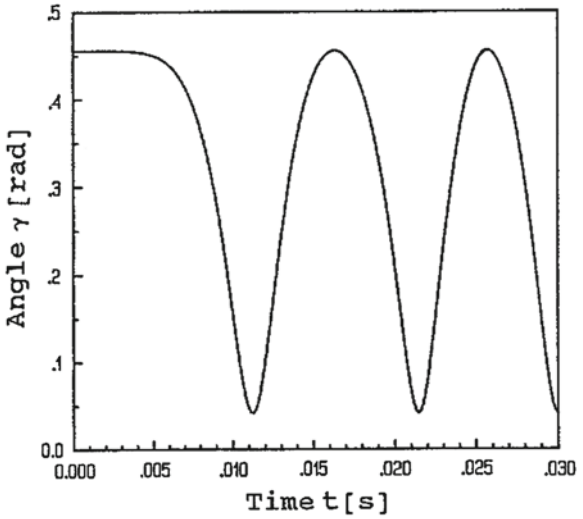


Fig. 15.5 Simulation of motion of mechanism [8]

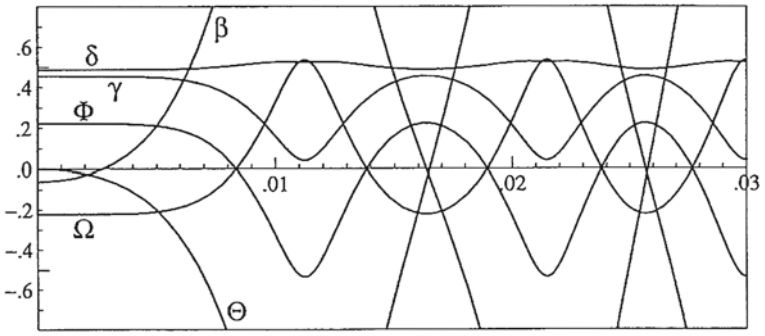


Fig. 15.6 Simulation of motion of mechanism [9]

	VAMPIRE	GENSYS	SIMPACK	ADAMS	NUCARS
Critical speed /m/s					
Vehicle 1	74	77.05	70	72	79
Vehicle 2	58	70.5	80	75	79

Fig. 15.7 Critical speeds [11]

as a benchmark problem. The parameters of the Iltis vehicle made under Volkswagen licence by Bombardier in Canada are described in detail including geometry and masses, force elements and a tire model. The vehicle has 4 identical suspensions with 1 degree of freedom each. Thus, the vehicle has 10 degrees of freedom altogether. Four test cases are identified: static equilibrium, eigenvalues, response to vertical road profiles and handling performance.

Due to four different codes, FASIM, MEDYNA, NEWEUL and SIMPACK many useful results are available. A 2 mm steering rack displacement with a ramp-to-step input results after transition in a steady-state circular cornering. SIMPACK and NEWEUL simulations coincide very well. For speeds of 10, 20 and 30 m/s lateral accelerations of 0.5, 1.5 and 2.5 m/s^2 are achieved with a maximum of 10 % overshoot.

Later in 1999 Iwnicki [11] edited the proceedings of the International Workshop on Computer Simulation of Rail Vehicle Dynamics held at Manchester Metropolitan University. Two vehicles and four track cases were designed representing typical modeling and simulation tasks in railway engineering. The results are very well documented. The benchmark vehicle 1 is a general passenger coach with two bogies, simple primary suspension and vertical secondary suspension with linear damping. The benchmark vehicle 2 is a two axle freight car with load depending friction damping. The benchmark vehicle models run on four different track cases of different complexity. The contact between the wheels and the rails is not specified, the contact modeling is part of the software packages VAMPIRE, GENSYS, SIMPACK, ADAMS and NUCARS participating in the Manchester Benchmarks. Very important in railway engineering is the critical speed. Some results are shown in Fig. 15.7.

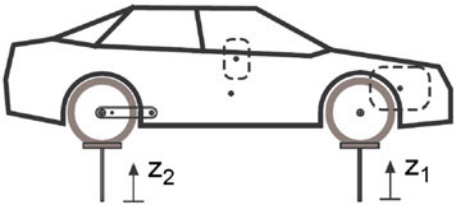


Fig. 15.8 Two-axle vehicle as test case [12, 13]

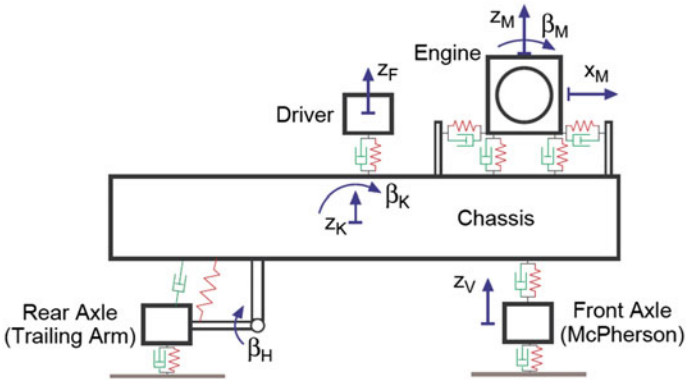


Fig. 15.9 Generalized coordinates of two-axle vehicle [12, 13]

In principle, multibody computer codes for simulations include modeling and time integration programs which may also considered separately. Rill and Schiehlen [12] used 2009 the bouncing and pitching motion of a two-axle vehicle for testing MATLAB time integrators, Fig. 15.8. There were compared nine different codes subject to bump and random road excitation. The assessment criteria are efficiency and accuracy.

The benchmark parameters characterizing inertia, geometry, suspension stiffness and damping as well as the tire for a medium size passenger car are listed in detail in Rill [13]. Then, the nonlinear equation of motion are generated for the system with 8 degrees of freedom, Fig. 15.9, and transformed to state equations which are solved by MATLAB solvers [14], Table 15.2. The excitations by the guideway $z_1(t)$ and $z_2(t)$ are assumed to be a deterministic cosine-shaped bump or a completely random uneven road, respectively. For the validation of the benchmark linear motions featuring eigenvalue analyses are used where the state equations are linearized with respect to the equilibrium position. The eigenfrequencies found with the benchmark parameters are typical for a medium-size passenger car.

Since analytical solutions are not available, for the reference a partially implicit Euler solver denoted as ode1m is applied to the equations of motion, too. For this benchmark the vertical vibrations of a planar vehicle model have been used for the

Table 15.2 Solvers for ordinary differential equations [14]

Solver	Problem type	Order of accuracy	When to use
`ode45`	Nonstiff	Medium	Most of the time. This should be the first solver you try
`ode23`	Nonstiff	Low	For problems with crude error tolerances or for solving moderately stiff problems
`ode113`	Nonstiff	Low to high	For problems with stringent error tolerances or for solving computationally intensive problems
`ode15s`	Stiff	Low to medium	If `ode45` is slow because the problem is stiff
`ode23s`	Stiff	Low	If using crude error tolerances to solve stiff systems and the mass matrix is constant
`ode23t`	Moderately Stiff	Low	For moderately stiff problems if you need a solution without numerical damping
`ode23tb`	Stiff	Low	If using crude error tolerances to solve stiff systems

assessment. For both excitations, bump and uneven road, the Matlab ode45 solver, a one-step solver. based on an explicit Runge-Kutta (4,5) formula, was the best one.

15.2.4 Flexible Multibody Systems

Recently a fourth class of benchmarks, flexible multibody systems, is considered by Bauchau [15]. The benchmarks deal with a four-bar mechanism and the lateral buckling of a moving beam. Benchmark test cases are available for different types of structural elements often found in flexible multibody dynamics codes. Each benchmark problem is presented using a uniform template that presents the following data: description of the problem, input file in html format, complete input file containing all the input data in plain text format, detailed numerical results in plain text format, and movies. These results can be imported in GNU Octave for plotting and comparison with other predictions.

Figure 15.10 defines a flexible planar four bar mechanism. In the reference configuration, the bars of this planar mechanism intersect each other at 90° angles and the axes of rotation of the revolute joints at points **A**, **B**, and **D** are normal to the plane of the mechanism. However, the axis of rotation of the revolute joint at point **C** is misaligned at a 5° angle with respect to this normal to simulate an initial defect in the mechanism. The angular velocity at point **A** of bar 1 is prescribed to be $\Omega = 5$ rad/s.

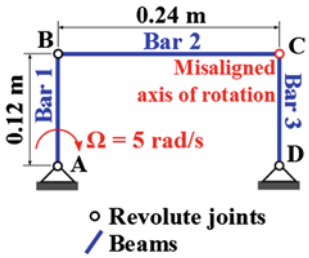


Fig. 15.10 Flexible mechanism [15]

Bars 1 and 2 are of square cross-section of size 16 by 16 mm; bar 3 has a square cross-section of size 8 by 8 mm. The three bars are made of steel, whose mechanical characteristics are Young's modulus E = 207 GPa and Poisson's ratio $\nu = 0.3$.

If the bars were infinitely rigid, no motion would be possible because the mechanism locks. For elastic bars, motion becomes possible, but generates large, rapidly varying internal forces and moments. This problem was simulated for a total of 12 s using 3,000 time steps of constant size $\Delta t = 4$ ms. If the four revolute joints had their axes of rotation orthogonal to the plane of the mechanism, the response of the system would be purely planar, and bars 1 and 3 would rotate at constant angular velocities around points **A** and **D**, respectively. The initial defect in the flexible mechanism causes a markedly different response. Bar 1 rotates at the constant prescribed angular velocity, but bar 3 now oscillates back and forth, never completing an entire turn.

15.3 Library of Benchmark Problems

A list of 27 entrees of software for multibody system simulations is provided by McPhee [16]. Many of them include also problems which may be used as further test cases. In 2012 the IFToMM Technical Committee for Multibody Dynamics started a project on multibody benchmarks. In a first step, a double four-bar mechanism was considered by the research groups of McPhee [16] and Cuadrado [17], see also Fig. 15.11. In a second step, a special session on benchmark problems was organized at the ECCOMAS Multibody Dynamics Conference 2013. Four presentations have been delivered by Bauchau [18], Masoudi et al. [19], Schiehlen [20], and Valasek and Sika [21]. Furthermore, a more standardized manner for the definition of benchmarks was discussed providing the following information.

- Description of the benchmark system with respect to the number of degrees of freedom, the elements selected (rigid and/or flexible bodies, springs, dampers, force or position actuators,. . .), and the coordinates chosen (minimal, non-minimal).
- Definitions and numbers of the parameters of the system related to the elements selected (inertia, spring and damper coefficients, nonlinear characteristics,. . .).
- Initial conditions, force actuation (direct dynamics) and position actuation (indirect dynamics).
- Time integration methods used and eigenfrequency analysis performed.

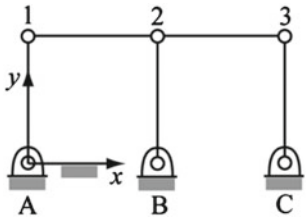


Fig. 15.11 Double four-bar mechanism

Table 15.3 Double four-bar mechanism submissions

Submitter	Accuracy (energy drift)	Efficiency (performance)	Coordinates
Javier Cuadrado University of A Coruña, Spain	0.0917	0.6000	Natural
Alberto Luaces University of A Coruña, Spain	0.0137	1.7000	Relative
Francisco Gonzalez McGill University, Canada	0.0917	0.1450	Natural
Markus Burkhardt University of Stuttgart, Germany	0.0015	0.0568	Minimal
Pierangelo Masarati Politecnico di Milano, Italy	0.0900	0.3250	Inertial

In the IFToMM Library of Computational Benchmark Problems for Multibody Dynamics [22] only problems should be published for which at least one group has submitted some results. However, it is most preferable to have two or more submissions for each problem. Furthermore, a small committee should be established to initiate and check the submissions before results are added to the Library.

First results are now available for the double four-bar mechanism, Table 15.3. From this problem featuring closed kinematical loops can be learnt that the modelling approach characterized by the coordinates chosen has a major influence an the accuracy and performance of the simulations. More effort with modelling may pay off even if sophisticated numerical codes like DAE solvers are used.

The biomechanical benchmark of the Library deals with the 2D inverse dynamic gait analysis of humans, and a 3D rigid slider-crank mechanism has been submitted to the Library, too.

Benchmarks with different complexity within one area of applications are also useful for the verification of simulation results as shown for the lateral dynamics of rail and road vehicles [23]. Using the same assumptions, the more simple models may serve as reliable benchmarks for the verification of computational results of higher complexity models.

15.4 Conclusion

Computational dynamics of multibody systems require the verification by benchmarks and/or experiments. While experimental research is expensive and time-consuming, the verification by benchmarks is an attractive alternative. For this purpose appropriate benchmarks have to be defined and tested. Library of Computational Benchmark Problems presents an open tool to develop and provide benchmarks. The Library website is intended to be a tool for the international multibody dynamics community to propose, solve, and refer to a collection of benchmark problems. Members of the community can view the results obtained by other researchers, submit their own results for others to reference, and even propose new benchmark problems that can help advance the state-of-the-art in multibody system dynamics.

References

1. http://en.wikipedia.org/wiki/Benchmark_(computing), September 2012
2. http://en.wikipedia.org/wiki/Test_suite, November 2012
3. Magnus K (1942) Über die Anwendungen der allgemeinen Bewegungsgleichungen starrer Körper in bewegten Bezugssystemen (in German). Z. angew. Math. Mech. 22:336–356
4. Magnus K (1966) Die Stabilität permanenter Drehungen des astatischen Kardankreisels mit unsymmetrischem Rotor (in German). Acta Mechanica 2:130–143
5. Magnus K (1971) Kreisel - Theorie und Anwendungen (in German). Springer, Berlin
6. Giles DRA (1978) A comparison of three problem-oriented simulation programs of dynamic mechanical systems. Univ. Waterloo, Ontario, Thesis
7. Manning DW (1981) A computer technique for simulating dynamic multibody systems based on dynamic formalism. Univ. Waterloo, Ontario, Thesis
8. Schiehlen W (1990) Multibody systems handbook. Springer, Berlin
9. E. Hairer and G. Wanner. Solving ordinary differential equations II, pages 530–540. Springer, Berlin, 1996.
10. Kortüm W, Sharp RS (1993) Multibody computer codes in vehicle dynamics. Swets & Zeitlinger, Amsterdam
11. Iwnicki S (1999) The Manchester benchmarks for rail vehicle simualtions. Swets & Zeitlinger, Amsterdam
12. G. Rill and W. Schiehlen. Performance assessment of time integration methods for vehicle dynamics simulations. In K. Arczewski, J. Frazek and M. Wojtyra, editors, Multibody Dynamics 2009, 9 pages (pdf), ECCOMAS Thematic Conference, Warsaw, 2009.
13. G. Rill. Vergleich von Integrationsverfahren am Beispiel eines ebenen PKW-Modells (in German). Institutsbericht IB-40, University of Stuttgart, Inst. Techn. Num. Mechanik, Stuttgart, 2008.
14. http://www.mathworks.de/de/help/matlab/ref/ode23.html, June 2013
15. O. A. Bauchau. http://www.dymoresolutions.com/Benchmarks/Benchmarks.html, June 2013
16. J. McPhee. http://real.uwaterloo.ca/~mbody/#Software, September 2012
17. J. Cuadrado. http://lim.ii.udc.es/people.en.html#Head, June 2013
18. Bauchau OA (2013) Benchmark problems for beam models in flexible multibody dynamics. In: Terze Z (ed) Multibody Dynamics 2013. Faculty Mech. Eng. Nav. Architecture, Zagreb, pp 493–494

19. Masoudi R, Uchida T, Vilela D, Luaces A, Cuadrado J, McPhee J (2013) A Library of computational benchmark problems. In: Terze Z (ed) Multibody Dynamics 2013. Faculty Mech. Eng. Nav. Architecture, Zagreb, pp 495–496
20. Schiehlen W (2013) History of benchmark problems in multibody dynamics. In: Terze Z (ed) Multibody Dynamics 2013. Faculty Mech. Eng. Nav. Architecture, Zagreb, pp 497–498
21. Valasek M, Sika Z (2013) Benchmark problems for computational efficiency of rigid multibody system dynamics. In: Terze Z (ed) Multibody Dynamics 2013. Faculty Mech. Eng. Nav. Architecture, Zagreb, pp 499–500
22. http://iftomm-multibody.org/benchmark/, January 2014
23. W. Schiehlen. On the history of lateral dynamics modelling for rail and road vehicles. In Proceedings of the ASME 2013 IDETC/CIE, Portland, OR, USA, August 4–7, 2013. ASME, New York, 2013, Paper DETC2013-12801, 9 pp. (pdf).