

Computational Methods in Applied Sciences

Josep M. Font-Llagunes *Editor*

Multibody Dynamics

Computational Methods and
Applications



 Springer

The Springer logo, which is a stylized chess knight piece.

Computational Methods in Applied Sciences

Volume 42

Series editor

E. Oñate
CIMNE
Edificio C-1, Campus Norte UPC
Gran Capitán, s/n.
08034 Barcelona, Spain
e-mail: onate@cimne.upc.edu

More information about this series at <http://www.springer.com/series/6899>

Josep M. Font-Llagunes
Editor

Multibody Dynamics

Computational Methods and Applications

 Springer

Editor

Josep M. Font-Llagunes
Department of Mechanical Engineering
Universitat Politècnica de Catalunya
Barcelona
Spain

ISSN 1871-3033

Computational Methods in Applied Sciences

ISBN 978-3-319-30612-4

ISBN 978-3-319-30614-8 (eBook)

DOI 10.1007/978-3-319-30614-8

Library of Congress Control Number: 2016934018

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG Switzerland

Preface

This book includes state-of-the-art contributions in the field of multibody dynamics, an area of computational mechanics aimed at studying the dynamic behaviour of mechanical systems composed of several rigid or flexible bodies that are connected to each other by joints and actuated by forces. Applications of multibody dynamics are related to many fields of contemporary engineering, such as vehicle and railway systems, aeronautical and space vehicles, robotic manipulators, mechatronic and autonomous systems, smart structures, biomechanical systems, and nanotechnologies.

Each book chapter contains a revised and extended version of a paper that was presented at the ECCOMAS Thematic Conference on Multibody Dynamics 2015 held in the Barcelona School of Industrial Engineering, Universitat Politècnica de Catalunya, on 29 June–2 July 2015. The selected works reflect the state of the art in the application of multibody dynamics methodologies to different topics, namely, formulations and numerical methods, efficient methods and real-time applications, flexible multibody dynamics, contact dynamics and constraints, multiphysics and coupled problems, control and optimization, software development and computer technology, aerospace and maritime applications, biomechanics, railroad vehicle dynamics, road vehicle dynamics, and robotics. The result is a comprehensive text that covers fundamental and applied topics, which can serve as a reference handbook for senior researchers, doctoral students and engineers who aim to apply multibody dynamics techniques to different fields of engineering and applied physics.

The ECCOMAS Thematic Conference on Multibody Dynamics is an international meeting held once every two years in a European country. Continuing the very successful series of past conferences that have been organized in Lisbon (2003), Madrid (2005), Milan (2007), Warsaw (2009), Brussels (2011) and Zagreb (2013), the 2015 edition organized in Barcelona served once again as a meeting point for the international researchers, scientists and experts from academia, research laboratories and industry working in the area of multibody dynamics.

The conference was organized by the Department of Mechanical Engineering of the Universitat Politècnica de Catalunya (UPC) in Barcelona, and brought together 360 participants from 35 countries spanning 5 continents. The total number of presentations was 281, which were grouped in 5 parallel tracks giving a total number of 65 sessions. Moreover, four keynote lectures were organized covering the areas of robotics, biomechanics, vehicle dynamics, and design and control of mechanical systems. Two awards to the best scientific contribution and the best work by a young researcher were given during the closing ceremony of the conference.

I would like to take this opportunity to thank the authors for submitting their excellent contributions, the keynote lecturers for accepting the invitation and for the quality of their outstanding talks, the awards and scientific committees for their support to the organization of the conference, the session organizers for reviewing all extended abstracts and selecting the awards nominees, and last but not least, the local organizing committee for all the hard work that made the conference possible. I am also grateful to ECCOMAS, the principal supporting institution; the other international supporting organizations, IFToMM, IUTAM, JSME and KSME; and Springer for accepting the publication of this book.

Barcelona
January 2016

Josep M. Font-Llagunes

Contents

1	Numerical Integration of Underactuated Mechanical Systems Subjected to Mixed Holonomic and Servo Constraints	1
	Peter Betsch, Robert Altmann and Yinping Yang	
2	Enhancing the Performance of the DCA When Forming and Solving the Equations of Motion for Multibody Systems	19
	Jeremy J. Laflin, Kurt S. Anderson and Mike Hans	
3	Three-Dimensional Non-linear Shell Theory for Flexible Multibody Dynamics	33
	Shilei Han and Olivier A. Bauchau	
4	On the Frictional Contacts in Multibody System Dynamics	67
	Filipe Marques, Paulo Flores and Hamid M. Lankarani	
5	Modeling and Simulation of a 3D Printer Based on a SCARA Mechanism	93
	Eduardo Paiva Okabe and Pierangelo Masarati	
6	Structure Preserving Optimal Control of a Three-Dimensional Upright Gait	115
	Michael W. Koch and Sigrid Leyendecker	
7	Robotran-YARP Interface: A Framework for Real-Time Controller Developments Based on Multibody Dynamics Simulations	147
	Timothée Habra, Houman Dallali, Alberto Cardellino, Lorenzo Natale, Nikolaos Tsagarakis, Paul Fisette and Renaud Ronsse	

8	Wheel-Ground Modeling in Planetary Exploration: From Unified Simulation Frameworks Towards Heterogeneous, Multi-tier Wheel Ground Contact Simulation.	165
	Roy Lichtenheldt, Stefan Barthelmes, Fabian Buse and Matthias Hellerer	
9	Intervention-Autonomous Underwater Vehicle Multibody Models for Dynamic Manipulation Tasks	193
	Roberto Conti, Riccardo Costanzi, Francesco Fanelli, Enrico Meli, Alessandro Ridolfi and Benedetto Allotta	
10	Development of a Musculotendon Model Within the Framework of Multibody Systems Dynamics	213
	Ana R. Oliveira, Sérgio B. Gonçalves, Mamede de Carvalho and Miguel T. Silva	
11	Numerical and Experimental Study on Contact Force Fluctuation Between Wheel and Rail Considering Rail Flexibility and Track Conditions.	239
	Saki Ienaga, Yoshiaki Terumichi, Kazuhiko Nishimura and Minoru Nishina	
12	Use of Flexible Models in Extended Kalman Filtering Applied to Vehicle Body Force Estimation.	259
	Sebastiaan van Aalst, Frank Naets, Johan Theunissen and Wim Desmet	
13	Design and Control of an Energy-Saving Robot Using Storage Elements and Reaction Wheels	277
	Makoto Iwamura, Shunichi Imafuku, Takahiro Kawamoto and Werner Schiehlen	
14	Exploiting the Equations of Motion For Biped Robot Control with Enhanced Stability	299
	Johannes Mayr, Alexander Reiter, Hubert Gattringer and Andreas Müller	

Chapter 1

Numerical Integration of Underactuated Mechanical Systems Subjected to Mixed Holonomic and Servo Constraints

Peter Betsch, Robert Altmann and Yinping Yang

Abstract A new index reduction approach is developed for the inverse dynamics simulation of underactuated mechanical systems. The underlying equations of motion contain both holonomic and servo constraints. The proposed method is applied to a very general and versatile formulation of cranes. The numerical results demonstrate the functional efficiency of the method.

Keywords Underactuated mechanical systems · Feedforward control · Inverse dynamics · Differentially flat systems

1.1 Introduction

We present a new approach to the inverse dynamics simulation of discrete mechanical systems. The proposed method is relying on the use of servo constraints for the partial specification of the motion of mechanical systems (see, for example, [1–3]). In particular, we focus on the specification of trajectories of specific points of a multibody system such as the end effector of a robot.

The partial specification of the motion of a multibody system by means of servo constraints typically leads to a problem formulation in terms of differential-algebraic equations (DAEs). If minimal coordinates are used, the differential part of the DAEs corresponds to the equations of motion while the algebraic part is related to the servo constraints. The servo constraints enforce the desired motion along prescribed

P. Betsch (✉) · Y. Yang
Institut für Mechanik, Karlsruher Institut für Technologie, Kaiserstraße 12,
76131 Karlsruhe, Germany
e-mail: peter.betsch@kit.edu

Y. Yang
e-mail: yinping.yang@kit.edu

R. Altmann
Institut für Mathematik, Technische Universität Berlin, StraÙes 17. Juni 136,
10623 Berlin, Germany
e-mail: raltmann@math.tu-berlin.de

trajectories and thus specify the control outputs of the system. To determine the associated control inputs required to steer the system such that the prescribed trajectories are tracked, the DAEs need to be solved. In this way, a simulation approach to the feedforward control of multibody systems can be realized.

In the special case of fully actuated multibody systems the simulation approach to the inverse dynamics problem yields index-3 DAEs that can be integrated in analogy to the DAEs corresponding to constrained mechanical systems (see, for example, [4]). However, the situation changes considerably if underactuated mechanical systems are dealt with. In this type of systems the number of degrees of freedom exceeds the number of controls. Examples of underactuated systems are cranes and flexible multibody systems. The use of servo constraints in the context of underactuated multibody systems leads to a broad diversity of servo constraint problems (see, in particular, the recent papers [5–7]). One indicator of problem diversity is the (differentiation) index of the underlying DAEs that typically ranges from three to five and even higher. Consequently, to facilitate a stable numerical integration some kind of index reduction approach needs to be applied.

In the present work we apply a specific index reduction technique called minimal extension (see [8]). Index reduction by minimal extension is based on the introduction of new algebraic variables along with the enlargement of the DAEs by appending time derivatives of the constraints. In our previous work [9] we have shown that index reduction by minimal extension can be applied very efficiently by exploiting the specific structure provided by underactuated mechanical systems. In this connection either minimal coordinates or redundant coordinates can be used. We verified that our approach is a viable alternative to the projection method developed in [10].

We focus on the inverse dynamics of a family of crane models that are known to belong to the class of differentially flat systems. We have shown in our previous work [9] that in a first step the minimal extension approach can be used to lower the index of the DAEs from five to three. In a second step the index can even be reduced to one.

The goal of the present work is to extend the applicability of our index reduction approach to mechanical models of underactuated systems that rely on arbitrarily selected redundant coordinates. Specifically, in contrast to our previous work [9], the number of holonomic constraints is not limited. Consequently, general crane formulations such as those developed in [11] can now be included into the present index reduction approach. Similarly, other rotationless formulations of multibody dynamics such as natural coordinates or Cosserat-type descriptions in terms of directors (including rigid bodies and nonlinear beams and shells) typically yield a large number of holonomic constraints. These formulations are now embraced as well by our newly developed index reduction method.

An outline of the rest of our paper is as follows. In Sect. 1.2 we introduce the general description of mechanical systems subjected to both holonomic and servo constraints. In Sect. 1.3 we present index reduction by minimal extension and link the present formulation to our previous work [9]. After the discretization in time of the present DAEs in Sect. 1.4, a sample application is dealt with in Sect. 1.5. Eventually, conclusions are drawn in Sect. 1.6.

1.2 Inverse Dynamics of Underactuated Mechanical Systems

We start with a general formulation of mechanical systems subjected to both holonomic and servo constraints. In particular, we consider equations of motion of the form

$$\begin{bmatrix} \mathbf{M}_1(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ \ddot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{p}, \dot{\mathbf{p}}) \\ \mathbf{f}_2(\mathbf{x}, \dot{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1^T(\mathbf{p}) \\ \mathbf{0} \end{bmatrix} \mathbf{u} - \mathbf{G}^T(\mathbf{p}, \mathbf{x})\boldsymbol{\lambda}, \quad (1.1a)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{p}), \quad (1.1b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{p}, \mathbf{x}), \quad (1.1c)$$

$$\mathbf{x} = \boldsymbol{\gamma}. \quad (1.1d)$$

The first row block in (1.1a) corresponds to the robot (or input) subsystem with coordinates $\mathbf{p} \in \mathbb{R}^{n-a}$, whereas the second row block in (1.1a) corresponds to the output subsystem with coordinates $\mathbf{x} \in \mathbb{R}^a$. The n redundant coordinates

$$\mathbf{q} = \begin{bmatrix} \mathbf{p} \\ \mathbf{x} \end{bmatrix} \quad (1.2)$$

are subject to the holonomic constraints (1.1b) and (1.1c), with associated constraint functions $\mathbf{h} \in \mathbb{R}^{m_1}$ and $\mathbf{g} \in \mathbb{R}^{m_2}$. The total number of holonomic constraints is denoted by $m = m_1 + m_2$. Note that the constraint function \mathbf{h} does not depend on the output variables \mathbf{x} . The Jacobian of the holonomic constraints assumes the form

$$\mathbf{G}(\mathbf{p}, \mathbf{x}) = \begin{bmatrix} \mathbf{H}_1(\mathbf{p}) & \mathbf{0} \\ \mathbf{G}_1(\mathbf{p}, \mathbf{x}) & \mathbf{G}_2(\mathbf{p}, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} \partial_{\mathbf{p}}\mathbf{h}(\mathbf{p}) & \mathbf{0} \\ \partial_{\mathbf{p}}\mathbf{g}(\mathbf{p}, \mathbf{x}) & \partial_{\mathbf{x}}\mathbf{g}(\mathbf{p}, \mathbf{x}) \end{bmatrix} \in \mathbb{R}^{m,n}. \quad (1.3)$$

The Lagrange multipliers associated to the m holonomic constraints are contained in

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} \in \mathbb{R}^m, \quad \text{with } \boldsymbol{\lambda}_1 \in \mathbb{R}^{m_1}, \boldsymbol{\lambda}_2 \in \mathbb{R}^{m_2}.$$

Due to the presence of holonomic constraints, the configuration space of the constrained mechanical system under consideration is defined by

$$\mathbf{Q} = \{\mathbf{q} \in \mathbb{R}^n \mid \mathbf{h}(\mathbf{p}) = \mathbf{0}, \mathbf{g}(\mathbf{p}, \mathbf{x}) = \mathbf{0}\} \quad (1.4)$$

Throughout this work we assume that the constraints are independent. Consequently, the constraint Jacobian \mathbf{G} has full row rank and the discrete mechanical system under consideration has $n - m$ degrees of freedom.

The servo constraints (1.1d) specify the desired trajectory of the load via the prescribed function $\boldsymbol{\gamma} : \mathbb{I} \rightarrow \mathbb{R}^a$, where $\mathbb{I} = [t_0, t_f]$ is the time interval of interest. In the present work we focus on underactuated mechanical systems in which the number of controls is smaller than the number of degrees of freedom, i.e., $a < n - m$.

The control inputs $\mathbf{u} \in \mathbb{R}^a$ regulate the control forces acting on the robot subsystem. In this connection $\mathbf{B}_1 \in \mathbb{R}^{a, n-a}$ denotes the input transformation matrix. Besides the constraint and control forces, additional forces acting on the system are contained in the conjugate force vectors $\mathbf{f}_1 \in \mathbb{R}^{n-a}$ and $\mathbf{f}_2 \in \mathbb{R}^a$. Similarly, the mass matrix is split into the submatrices $\mathbf{M}_1 \in \mathbb{R}^{n-a, n-a}$ and $\mathbf{M}_2 \in \mathbb{R}^{a, a}$.

Due to the presence of servo constraints the (differentiation) index of the DAEs (1) often exceeds 3. For example, the application to (differentially flat) crane systems typically yields an index of 5. Consequently, prior to the application of a numerical integrator the index of the DAEs should be lowered. For that purpose, following our previous work [9], we apply index reduction by minimal extension to the DAEs (1).

We emphasize at this point that in the above formulation the number of holonomic constraints, m , is just restricted by $m < n$. This facilitates the arbitrary selection of redundant coordinates best suited for the description and numerical simulation of the specific inverse dynamics problem at hand.

1.3 Index Reduction by Minimal Extension

Guided by our previous work [9] we enlarge the system of DAEs (1) by appending the first and second time derivative of the servo constraints. To maintain a square system we introduce additional dummy derivatives $\widehat{\mathbf{x}} := \dot{\mathbf{x}}$ and $\widetilde{\mathbf{x}} := \ddot{\mathbf{x}}$. Accordingly, we arrive at

$$\begin{bmatrix} \mathbf{M}_1(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ \widetilde{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{p}, \dot{\mathbf{p}}) \\ \mathbf{f}_2(\mathbf{x}, \widehat{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1^T(\mathbf{p}) \\ \mathbf{0} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{H}_1^T(\mathbf{p}) & \mathbf{G}_1^T(\mathbf{p}, \mathbf{x}) \\ \mathbf{0} & \mathbf{G}_2^T(\mathbf{p}, \mathbf{x}) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \quad (1.5a)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{p}), \quad (1.5b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{p}, \mathbf{x}), \quad (1.5c)$$

$$\mathbf{x} = \boldsymbol{\gamma}, \quad (1.5d)$$

$$\widehat{\mathbf{x}} = \dot{\boldsymbol{\gamma}}, \quad (1.5e)$$

$$\widetilde{\mathbf{x}} = \ddot{\boldsymbol{\gamma}}, \quad (1.5f)$$

Within this paper we show that—provided certain assumptions apply—the minimally extended system (5) has index 3. Typical applications are differentially flat crane models where the index equals 5 in the original form. In our previous work [9] this has been shown for the special case $m \leq a$ and $\mathbf{M}_1(\mathbf{p})$ non-singular. Here we allow that there are more holonomic constraints than servo constraints.

To guarantee the index-3 property of system (5), we state the following two assumptions. The first assumption ensures, amongst others, that the number of holonomic constraints depending on \mathbf{x} is bounded by the dimension of \mathbf{x} , namely a .

Assumption 1 The block $\mathbf{G}_2^T(\mathbf{p}, \boldsymbol{\gamma})$ of the Jacobian \mathbf{G} is of full rank and $m_2 \leq a$. Furthermore, the dimensions satisfy

$$2a + m_1 \leq n \quad \text{and} \quad a < n - m.$$

The last inequality ensures that the the system is underactuated. Note that the two previous assumptions already imply $a \leq n - m$.

Within the proof of Theorem 1 below, we reduce and rewrite the equations such that the resulting system is of index 3. This requires a certain matrix to be invertible which we summarize in the following assumption.

Assumption 2 If Assumption 1 is satisfied, then there exists a matrix $\mathbf{P}_2 \in \mathbb{R}^{a, a-m_2}$, whose columns span the null space of \mathbf{G}_2 , i.e., $\mathbf{G}_2(\mathbf{p}, \boldsymbol{\gamma})\mathbf{P}_2(\mathbf{p}, \boldsymbol{\gamma}) = \mathbf{0}$. We define $\mathbf{z} \in \mathbb{R}^{a-m_2}$ by

$$\mathbf{z}(\mathbf{p}, \boldsymbol{\gamma}) := \mathbf{P}_2^T(\mathbf{p}, \boldsymbol{\gamma})(\mathbf{f}_2(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) - \mathbf{M}_2\ddot{\boldsymbol{\gamma}}) \quad (1.6)$$

and denote its derivative with respect to \mathbf{p} by $\mathbf{Z}_1(\mathbf{p}, \boldsymbol{\gamma}) = \partial_{\mathbf{p}}\mathbf{z}(\mathbf{p}, \boldsymbol{\gamma})$. Then, with \mathbf{G}_1 and \mathbf{H}_1 from (1.3), we assume that the matrix

$$\begin{bmatrix} \mathbf{M}_1(\mathbf{p}) & -\mathbf{B}_1^T(\mathbf{p}) & \mathbf{H}_1^T(\mathbf{p}) \\ \mathbf{Z}_1(\mathbf{p}, \boldsymbol{\gamma}) & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_1(\mathbf{p}, \boldsymbol{\gamma}) & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1(\mathbf{p}) & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n+m_1, n+m_1} \quad (1.7)$$

is invertible. Note that we do not ask the matrix $\mathbf{M}_1(\mathbf{p})$ itself to be invertible.

With the two assumptions in hand we are able to formulate the following theorem.

Theorem 1 *Given the Assumptions 1 and 2, the extended system (5) represents DAEs with index 3.*

Proof As mentioned before, the idea of the proof is to reduce the DAEs (5) to a system which has a similar structure as a constrained multibody system. Since the variables \mathbf{x} , $\hat{\mathbf{x}}$, and $\tilde{\mathbf{x}}$ are directly given by the prescribed trajectory $\boldsymbol{\gamma}$ and its derivatives, they may be eliminated from the system equations. Consider the second part of equation (1.5a), namely

$$\mathbf{M}_2\ddot{\boldsymbol{\gamma}} = \mathbf{f}_2(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) - \mathbf{G}_2^T(\mathbf{p}, \boldsymbol{\gamma})\boldsymbol{\lambda}_2.$$

The full rank property of \mathbf{G}_2 together with $m_2 \leq a$ from Assumption 1 then implies that we can solve for $\boldsymbol{\lambda}_2$ by

$$\boldsymbol{\lambda}_2 = (\mathbf{G}_2(\mathbf{p}, \boldsymbol{\gamma})\mathbf{G}_2^T(\mathbf{p}, \boldsymbol{\gamma}))^{-1} \mathbf{G}_2(\mathbf{p}, \boldsymbol{\gamma}) (\mathbf{f}_2(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) - \mathbf{M}_2\ddot{\boldsymbol{\gamma}}).$$

With the matrix $\mathbf{P}_2 \in \mathbb{R}^{a, a-m_2}$ from Assumption 2, which spans the null space of \mathbf{G}_2 , we define $\mathbf{z}(\mathbf{p}, \boldsymbol{\gamma}) \in \mathbb{R}^{a-m_2}$ as in (1.6). Note that this defines an algebraic constraint, since it has to be satisfied

$$\mathbf{z}(\mathbf{p}, \boldsymbol{\gamma}) = 0.$$

In summary, the remaining variables have to satisfy the system

$$\mathbf{M}_1(\mathbf{p})\ddot{\mathbf{p}} = \overline{\mathbf{f}}_1(\mathbf{p}, \dot{\mathbf{p}}) + \mathbf{B}_1^T(\mathbf{p})\mathbf{u} - \mathbf{H}_1^T(\mathbf{p})\boldsymbol{\lambda}_1, \quad (1.8a)$$

$$\mathbf{0} = \mathbf{z}(\mathbf{p}, \boldsymbol{\gamma}), \quad (1.8b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{p}, \boldsymbol{\gamma}), \quad (1.8c)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{p}) \quad (1.8d)$$

with

$$\overline{\mathbf{f}}_1(\mathbf{p}, \dot{\mathbf{p}}) := \mathbf{f}_1(\mathbf{p}, \dot{\mathbf{p}}) - \mathbf{G}_1^T(\mathbf{p}, \boldsymbol{\gamma})\boldsymbol{\lambda}_2 = \mathbf{f}_1 - \mathbf{G}_1^T(\mathbf{G}_2\mathbf{G}_2^T)^{-1}\mathbf{G}_2(\mathbf{f}_2 - \mathbf{M}_2\ddot{\mathbf{y}}).$$

System (8) consists of $n - a$ dynamic equations and $m_2 + m_1 + (a - m_2) = a + m_1$ constraints. Replacing the three constraints by their second derivatives, we can write system (8) in the form

$$\hat{\mathbf{M}} \begin{bmatrix} \ddot{\mathbf{p}} \\ \mathbf{u} \\ \boldsymbol{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{f}}_1(\mathbf{p}, \dot{\mathbf{p}}) \\ \mathbf{z}_2(\mathbf{p}, \dot{\mathbf{p}}, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}, \ddot{\boldsymbol{\gamma}}) \\ \mathbf{g}_2(\mathbf{p}, \dot{\mathbf{p}}, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}, \ddot{\boldsymbol{\gamma}}) \\ \mathbf{h}_2(\mathbf{p}, \dot{\mathbf{p}}) \end{bmatrix}$$

Therein, $\hat{\mathbf{M}}$ denotes the matrix from (1.7) and \mathbf{z}_2 , \mathbf{g}_2 , and \mathbf{h}_2 are vector-valued functions which include the remaining terms of the differentiation. Since $\hat{\mathbf{M}}$ is non-singular by Assumption 2, a multiplication by its inverse from the left yields an ODE for \mathbf{p} and algebraic equations for \mathbf{u} and $\boldsymbol{\lambda}_1$. Since only two differentiations were necessary, system (8) and thus, also system (5) have to be (at most) of index 3. \square

1.3.1 Reduction of the Number of Redundant Coordinates

We next connect the present formulation to our previous work [9]. To this end we eliminate the holonomic constraints (1.5b) by reducing the number of redundant coordinates from n to $\bar{n} = n - m_1$. This is possible if a mapping $\boldsymbol{\varphi} : \mathbb{R}^{\bar{n}-a} \rightarrow \mathbb{R}^{n-a}$ can be found such that

$$\mathbf{p} = \boldsymbol{\varphi}(\overline{\mathbf{p}}) \quad (1.9)$$

where $\overline{\mathbf{p}} \in \mathbb{R}^{\bar{n}-a}$ denotes the new redundant coordinates. The mapping (1.9) has to satisfy the constraints (1.5b) identically for arbitrary $\overline{\mathbf{p}} \in \mathbb{R}^{\bar{n}-a}$. Consequently,

$$\mathbf{h}(\boldsymbol{\varphi}(\bar{\mathbf{p}})) = \mathbf{0} \quad \text{and} \quad \mathbf{H}_1(\mathbf{p})D\boldsymbol{\varphi}(\bar{\mathbf{p}}) = \mathbf{0} \quad (1.10)$$

for $\mathbf{p} = \boldsymbol{\varphi}(\bar{\mathbf{p}})$. Premultiplying the first row block in (1.5a) by $D\boldsymbol{\varphi}^T(\bar{\mathbf{p}})$ and taking into account (1.9) and (1.10) yields the size-reduced DAEs

$$\begin{bmatrix} \bar{\mathbf{M}}_1(\bar{\mathbf{p}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix} \begin{bmatrix} \ddot{\bar{\mathbf{p}}} \\ \ddot{\tilde{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{f}}_1(\bar{\mathbf{p}}, \dot{\bar{\mathbf{p}}}) \\ \mathbf{f}_2(\mathbf{x}, \hat{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_1^T(\bar{\mathbf{p}}) \\ \mathbf{0} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \bar{\mathbf{G}}_1^T(\bar{\mathbf{p}}, \mathbf{x}) \\ \bar{\mathbf{G}}_2^T(\bar{\mathbf{p}}, \mathbf{x}) \end{bmatrix} \boldsymbol{\lambda}_2, \quad (1.11a)$$

$$\mathbf{0} = \bar{\mathbf{g}}(\bar{\mathbf{p}}, \mathbf{x}), \quad (1.11b)$$

$$\mathbf{x} = \boldsymbol{\gamma}, \quad (1.11c)$$

$$\hat{\mathbf{x}} = \dot{\boldsymbol{\gamma}}, \quad (1.11d)$$

$$\tilde{\mathbf{x}} = \ddot{\boldsymbol{\gamma}}, \quad (1.11e)$$

where

$$\bar{\mathbf{M}}_1(\bar{\mathbf{p}}) = D\boldsymbol{\varphi}^T(\bar{\mathbf{p}})\mathbf{M}_1(\mathbf{p})D\boldsymbol{\varphi}(\bar{\mathbf{p}}), \quad (1.12a)$$

$$\bar{\mathbf{f}}_1(\bar{\mathbf{p}}, \dot{\bar{\mathbf{p}}}) = D\boldsymbol{\varphi}^T(\bar{\mathbf{p}}) \left(\mathbf{f}_1(\mathbf{p}, \dot{\mathbf{p}}) - \mathbf{M}_1(\mathbf{p}) \frac{d}{dt} (D\boldsymbol{\varphi}(\bar{\mathbf{p}})) \dot{\bar{\mathbf{p}}} \right), \quad (1.12b)$$

$$\bar{\mathbf{B}}_1^T(\bar{\mathbf{p}}) = D\boldsymbol{\varphi}^T(\bar{\mathbf{p}})\mathbf{B}_1^T(\mathbf{p}), \quad (1.12c)$$

$$\bar{\mathbf{g}}(\bar{\mathbf{p}}, \mathbf{x}) = \mathbf{g}(\mathbf{p}, \mathbf{x}), \quad (1.12d)$$

$$\bar{\mathbf{G}}_1(\bar{\mathbf{p}}, \mathbf{x}) = \mathbf{G}_1(\mathbf{p}, \mathbf{x})D\boldsymbol{\varphi}(\bar{\mathbf{p}}), \quad (1.12e)$$

$$\bar{\mathbf{G}}_2(\bar{\mathbf{p}}, \mathbf{x}) = \partial_{\mathbf{x}}\bar{\mathbf{g}}(\bar{\mathbf{p}}, \mathbf{x}) \quad (1.12f)$$

for $\mathbf{p} = \boldsymbol{\varphi}(\bar{\mathbf{p}})$. In this way the number of redundant coordinates is reduced by m_1 such that the remaining coordinates are given by the \bar{n} -dimensional configuration vector

$$\bar{\mathbf{q}} = \begin{bmatrix} \bar{\mathbf{p}} \\ \mathbf{x} \end{bmatrix} \quad (1.13)$$

Note that (1.11b) contains the m_2 remaining holonomic constraints with associated Lagrange multipliers $\boldsymbol{\lambda}_2 \in \mathbb{R}^{m_2}$ in (1.11a). The configuration space of the constrained mechanical system under consideration can now be expressed in the form

$$\mathbf{Q} = \{\bar{\mathbf{q}} \in \mathbb{R}^{\bar{n}} \mid \bar{\mathbf{g}}(\bar{\mathbf{p}}, \mathbf{x}) = \mathbf{0}\} \quad (1.14)$$

The DAEs (11) lie at the heart of our previous work [9]. There it is shown that under certain conditions (e.g. $m_2 \leq a$ and $\bar{\mathbf{M}}_1(\bar{\mathbf{p}})$ non-singular) the minimally extended DAEs (11) attain an index reduction by two. In the case of differentially flat crane models the original DAEs have index 5, whereas the index-reduced DAEs (11) (and, correspondingly, the DAEs (5) as well) have index 3. It is further shown in [9] that a second application of index reduction by minimal extension can achieve a reduction to index-1 DAEs.

1.4 Discretization

For the specific inverse dynamics problems dealt with in the next section the proposed index reduction approach yields DAEs (5) with index 3. Due to the semi-explicit form of the DAEs (5) we can expect the simple Euler-backward discretization to work well (see Ascher & Petzold [12, Sect. 10.1.1]). The DAEs (5) can be recast in the form

$$\mathbf{M}_1(\mathbf{p})\ddot{\mathbf{p}} = \mathbf{f}_1(\mathbf{p}, \dot{\mathbf{p}}) + \mathbf{B}_1^T(\mathbf{p})\mathbf{u} - \mathbf{H}_1^T(\mathbf{p})\boldsymbol{\lambda}_1 - \mathbf{G}_1^T(\mathbf{p}, \boldsymbol{\gamma})\boldsymbol{\lambda}_2 \quad (1.15a)$$

$$\mathbf{0} = \mathbf{M}_2\ddot{\boldsymbol{\gamma}} - \mathbf{f}_2(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) + \mathbf{G}_2^T(\mathbf{p}, \boldsymbol{\gamma})\boldsymbol{\lambda}_2 \quad (1.15b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{p}, \boldsymbol{\gamma}) \quad (1.15c)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{p}) \quad (1.15d)$$

The DAEs (15) provide $n - a$ differential equations (1.15a) along with $a + m$ algebraic equations (1.15b) through (1.15d) for the determination of $\mathbf{p} \in \mathbb{R}^{n-a}$, $\mathbf{u} \in \mathbb{R}^a$, and $\boldsymbol{\lambda} \in \mathbb{R}^m$. Application of the Euler-backward method yields

$$\mathbf{p}_{n+1} - \mathbf{p}_n = \Delta t \mathbf{v}_{n+1} \quad (1.16a)$$

$$\begin{aligned} \mathbf{M}_1(\mathbf{p}_{n+1}) \frac{(\mathbf{v}_{n+1} - \mathbf{v}_n)}{\Delta t} &= \mathbf{f}_1(\mathbf{p}_{n+1}, \mathbf{v}_{n+1}) + \mathbf{B}_1^T(\mathbf{p}_{n+1})\mathbf{u}_{n+1} - \mathbf{H}_1^T(\mathbf{p}_{n+1})\boldsymbol{\lambda}_{1,n+1} \\ &\quad - \mathbf{G}_1^T(\mathbf{p}_{n+1}, \boldsymbol{\gamma}(t_{n+1}))\boldsymbol{\lambda}_{2,n+1} \end{aligned} \quad (1.16b)$$

$$\mathbf{0} = \mathbf{M}_2\ddot{\boldsymbol{\gamma}}(t_{n+1}) - \mathbf{f}_2(\boldsymbol{\gamma}(t_{n+1}), \dot{\boldsymbol{\gamma}}(t_{n+1})) + \mathbf{G}_2^T(\mathbf{p}_{n+1}, \boldsymbol{\gamma}(t_{n+1}))\boldsymbol{\lambda}_{2,n+1} \quad (1.16c)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{p}_{n+1}, \boldsymbol{\gamma}(t_{n+1})) \quad (1.16d)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{p}_{n+1}) \quad (1.16e)$$

In a typical time step of size $\Delta t = t_{n+1} - t_n$ we seek approximations $(\bullet)_{n+1}$ to $(\bullet)(t_{n+1})$ given the corresponding quantities $(\bullet)_n$ as result of the previous step. For the initial time step we require consistent initial values \mathbf{p}_0 and \mathbf{v}_0 that have to satisfy $\mathbf{g}(\mathbf{p}_0, \boldsymbol{\gamma}(t_0)) = \mathbf{0}$ and $\mathbf{h}(\mathbf{p}_0) = \mathbf{0}$ along with

$$\partial_{\mathbf{p}}\mathbf{g}(\mathbf{p}_0, \boldsymbol{\gamma}(t_0))\mathbf{v}_0 + \partial_{\mathbf{x}}\mathbf{g}(\mathbf{p}_0, \boldsymbol{\gamma}(t_0))\dot{\boldsymbol{\gamma}}(t_0) = \mathbf{0} \quad (1.17a)$$

$$\partial_{\mathbf{p}}\mathbf{h}(\mathbf{p}_0)\mathbf{v}_0 = \mathbf{0} \quad (1.17b)$$

The scheme (16) provides $2n + m - a$ algebraic equations for the determination of \mathbf{p}_{n+1} , $\mathbf{v}_{n+1} \in \mathbb{R}^{n-a}$, $\mathbf{u}_{n+1} \in \mathbb{R}^a$, and $\boldsymbol{\lambda}_{n+1} \in \mathbb{R}^m$.

1.5 Sample Application: 3D Rotary Crane

We demonstrate the present approach with the inverse dynamics simulation of a three-dimensional rotary crane. Previously, the crane under consideration has been formulated in terms of 6 redundant coordinates (see [9, 13]) or 5 minimal coordinates.

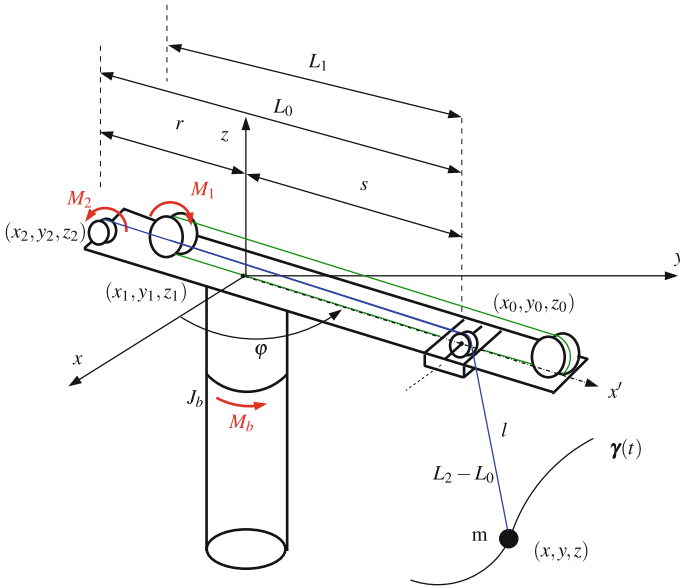


Fig. 1.1 The three-dimensional rotary crane in terms of $n = 10$ redundant coordinates

dinates (see [9, 14]). Alternatively, we now make use of the much more general framework for the modeling of cranes in [11]. For this, we use $n = 10$ redundant coordinates subjected to $m = 5$ holonomic constraints. The enlarged set of redundant crane coordinates, see Fig. 1.1, is given by

$$\mathbf{p} = [x_2 \ y_2 \ x_0 \ y_0 \ L_1 \ L_2 \ L_0]^T \tag{1.18}$$

and

$$\mathbf{x} = [x \ y \ z]^T \tag{1.19}$$

The last equation specifies the load (mass m) coordinates relative to a Cartesian inertial frame. The load is connected to the hoisting winch 2 (Cartesian coordinates $x_2, y_2, z_2 = 0$, actuating torque M_2 , radius r_2 , moment of inertia J_2) via a rope of length L_2 . The position of the trolley (Cartesian coordinates $x_0, y_0, z_0 = 0$, mass m_0) on the girder relative to the hoisting winch is given by L_0 . The trolley contains a pulley (radius r_w , moment of inertia J_w) and is moved along the girder under the action of a second winch 1 (Cartesian coordinates $x_1, y_1, z_1 = 0$, actuating torque M_1 , radius r_1 , moment of inertia J_1) whose position on the girder relative to the hoisting winch 2 is fixed by the parameter $\alpha = \frac{1}{2}$. The distance between winch 1 and the trolley is given by L_1 . The holonomic constraints $\mathbf{h}(\mathbf{p}) = \mathbf{0}$ are given by

and

$$\mathbf{H}_1^T = \begin{bmatrix} \alpha(x_0 - \alpha x_2) & x_0 - x_2 & -x_2 & -y_0 \\ \alpha(y_0 - \alpha_1 y_2) & y_0 - y_2 & -y_2 & x_0 \\ \alpha x_2 - x_0 & x_2 - x_0 & 0 & y_2 \\ \alpha y_2 - y_0 & y_2 - y_0 & 0 & -x_2 \\ L_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & L_0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_1^T = \begin{bmatrix} 0 \\ 0 \\ x - x_0 \\ y - y_0 \\ 0 \\ L_2 - L_0 \\ L_0 - L_2 \end{bmatrix}, \quad \mathbf{G}_2^T = \begin{bmatrix} x_0 - x \\ y_0 - y \\ -z \end{bmatrix}$$

Note that we have $a = 3$ control inputs given by the two winch torques M_1 , M_2 , along with the torque M_b acting about the z -axis of the rotary crane.

1.5.1 Check of the Assumptions

In order to show that the 3D rotary crane with redundant variables fits into the given framework, we need to check whether Assumptions 1 and 2 are satisfied.

Clearly \mathbf{G}_2^T is of full rank if either $x \neq x_0$, $y \neq y_0$, or $z \neq 0$. Note that this is a reasonable assumption since otherwise the position of the trolley would be equal to the position of the load. Furthermore, the dimensions satisfy $n = 10$, $m_1 = 4$, $m_2 = 1$, and $a = 3$ such that

$$1 = m_2 \leq a = 3, \quad 10 = 2a + m_1 \leq n = 10, \quad 3 = a < n - m = 5.$$

For the second assumption, we need $\mathbf{P}_2 \in \mathbb{R}^{a, a-m_2}$ which spans the null space of \mathbf{G}_2 . Depending on the case whether $x \neq x_0$, $y \neq y_0$, or $z \neq 0$, the projection \mathbf{P}_2 may be given by

$$\mathbf{P}_2 = \begin{bmatrix} y_0 - y & z \\ x - x_0 & 0 \\ 0 & x_0 - x \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} y_0 - y & 0 \\ x - x_0 & z \\ 0 & y_0 - y \end{bmatrix}, \quad \text{or} \quad \mathbf{P}_2 = \begin{bmatrix} z & 0 \\ 0 & z \\ x_0 - x & y_0 - y \end{bmatrix}.$$

W.l.o.g. we assume in the sequel that $z \neq 0$ which leads to

$$\mathbf{z} = \mathbf{P}_2^T(\mathbf{p}, \boldsymbol{\gamma})(f_2(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) - \mathbf{M}_2 \ddot{\boldsymbol{\gamma}}) = -m g \begin{bmatrix} x_0 - x \\ y_0 - y \end{bmatrix} - m \begin{bmatrix} z \ddot{\gamma}_1 + (x_0 - x) \ddot{\gamma}_3 \\ z \ddot{\gamma}_2 + (y_0 - y) \ddot{\gamma}_3 \end{bmatrix}.$$

and thus,

$$\mathbf{Z}_1(\mathbf{p}, \boldsymbol{\gamma}) = \partial_{\mathbf{p}} \mathbf{z}(\mathbf{p}, \boldsymbol{\gamma}) = \begin{bmatrix} 0 & 0 & -m(g + \ddot{\gamma}_3) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -m(g + \ddot{\gamma}_3) & 0 & 0 & 0 \end{bmatrix}.$$

Since in this special case the matrices

$$[-\mathbf{B}_1^T(\mathbf{p}) \mathbf{H}_1^T(\mathbf{p})] \quad \text{and} \quad [\mathbf{Z}_1^T(\mathbf{p}, \boldsymbol{\gamma}) \mathbf{G}_1^T(\mathbf{p}, \boldsymbol{\gamma}) \mathbf{H}_1^T(\mathbf{p})]$$

are square, it is sufficient to show the invertibility of the two matrices in order to prove that the matrix in (1.7) is invertible and thus, Assumption 2 is satisfied. A close look at the matrices then shows that minimal extension reduces the system equations of the 3D rotary crane to index 3 if the following conditions are satisfied:

$$z, L_0, L_1, L_2 \neq 0, \quad L_0 \neq L_2, \quad g \neq -\ddot{\gamma}_3, \quad x_0 x_2 + y_0 y_2 \neq 0, \quad x_2^2 + y_2^2 \neq 0.$$

These conditions are met by realistic applications of the rotary crane.

1.5.2 Reduction of the Number of Redundant Coordinates

We next link the above formulation of the rotary crane to the original one (see [9, 13]) relying on the reduced set of crane coordinates (Fig. 1.2)

$$\bar{\mathbf{p}} = [\varphi \ s \ l]^T \tag{1.20}$$

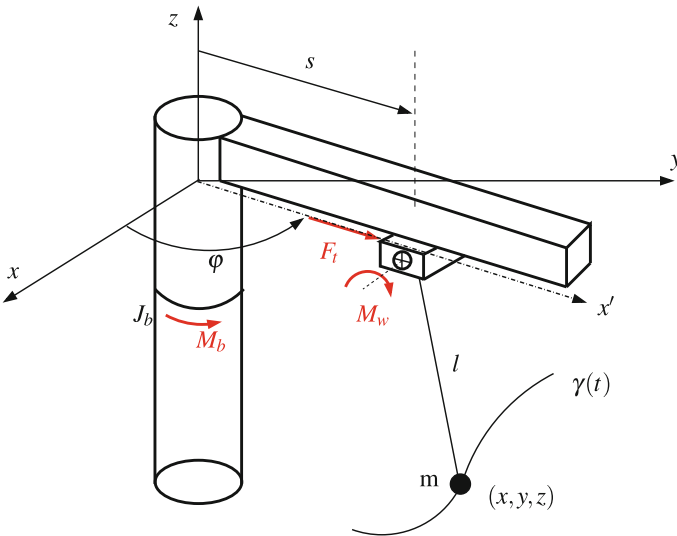


Fig. 1.2 The three-dimensional rotary crane in terms of a reduced set of $n = 6$ redundant coordinates

Here, the angle φ measures the rotation of the girder about the z-axis relative to the x-axis, s specifies the position of the trolley on the girder, and l denotes the length of the hoisting rope connecting the load with the winch contained in the trolley. In contrast to the previous crane model the winch contained in the trolley is now assumed to be actuated (torque M_w). The previous crane coordinates \mathbf{p} in (1.18) can now be expressed in terms of the reduced set of crane coordinates (1.20) giving rise to the mapping $\mathbf{p} = \boldsymbol{\varphi}(\bar{\mathbf{p}})$ in (1.9). Accordingly,

$$\begin{bmatrix} x_2 \\ y_2 \\ x_0 \\ y_0 \\ L_1 \\ L_2 \\ L_0 \end{bmatrix} = \begin{bmatrix} -r \cos \varphi \\ -r \sin \varphi \\ s \cos \varphi \\ s \sin \varphi \\ s + \alpha r \\ s + r + l \\ s + r \end{bmatrix} \quad \text{and} \quad D\boldsymbol{\varphi}(\mathbf{p}) = \begin{bmatrix} r \sin \varphi & 0 & 0 \\ -r \cos \varphi & 0 & 0 \\ -s \sin \varphi & \cos \varphi & 0 \\ s \cos \varphi & \sin \varphi & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Furthermore, the quantities in (12) can now be calculated in a straightforward way leading to

$$\bar{\mathbf{M}}_1(\bar{\mathbf{p}}) = \begin{bmatrix} Mr^2 + m_0s^2 & 0 & 0 \\ 0 & m_0 + \frac{J_1}{r_1^2} + \frac{J_2}{r_2^2} & \frac{J_2}{r_2^2} \\ 0 & \frac{J_2}{r_2^2} & \frac{J_2}{r_2^2} + \frac{J_w}{r_w^2} \end{bmatrix}$$

and

$$\bar{\mathbf{f}}_1(\bar{\mathbf{p}}, \dot{\bar{\mathbf{p}}}) = \begin{bmatrix} -2m_0s\dot{s}\dot{\varphi} \\ m_0s\dot{\varphi}^2 \\ 0 \end{bmatrix}, \quad \bar{\mathbf{B}}_1^T(\bar{\mathbf{p}}) = \begin{bmatrix} 0 & 0 & 1 \\ \frac{1}{r_1} & \frac{1}{r_2} & 0 \\ 0 & \frac{1}{r_2} & 0 \end{bmatrix}.$$

There remains one holonomic constraint (1.11b) which is given by

$$\bar{g}(\bar{\mathbf{p}}, \mathbf{x}) = \frac{1}{2}((x - s \cos \varphi)^2 + (y - s \sin \varphi)^2 + z^2 - l^2) = 0.$$

We finally remark that the control inputs $\bar{\mathbf{u}} := [M_b, F_t, M_w/r_w]^T$ conjugated to the reduced crane coordinates (1.20) can be obtained from $\bar{\mathbf{u}} = \bar{\mathbf{B}}_1^T(\bar{\mathbf{p}})\mathbf{u}$. In particular,

$$\begin{aligned} F_t &= \frac{M_1}{r_1} + \frac{M_2}{r_2} \\ M_w &= \frac{r_w}{r_2} M_2 \end{aligned}$$

That is, the two winch torques M_1 and M_2 of the original model are linked to the force F_t acting on the trolley and the winch torque M_w (cf. Fig. 1.2).

1.5.3 Numerical Results

In the numerical simulation we make use of the same data as in [9, 14]. In particular, the inertia parameters are given by $m = 100$, $m_0 = 10$, $J_b = 480$, and $M = \frac{J_b}{r^2}$ with $r = 4$. Concerning the moment of inertia corresponding to the winches, we choose $J_w = 0.1$, and $J_1 = J_2 = 0$. Moreover, $r_w = 0.1$. The servo constraints are used to prescribe a rest-to-rest maneuver of the load specified by $\boldsymbol{\gamma}(t) = \boldsymbol{\gamma}_0 + (\boldsymbol{\gamma}_f - \boldsymbol{\gamma}_0)c(t)$ with $\boldsymbol{\gamma}_0 = [5 \ 0 \ -5]^T$ at $t_0 = 0$ and $\boldsymbol{\gamma}_f = [-2 \ 2 \ -2]^T$ at $t_f = 20$. The function $c(t)$ is composed of three phases,

$$c(t) = \begin{cases} c_I(t) & \text{for } 0 \leq t < 5, \\ c_{II}(t) & \text{for } 5 \leq t < 15, \\ c_{III}(t) & \text{for } 15 \leq t \leq 20 \end{cases}$$

with

$$\begin{aligned} c_I(t) &= \frac{1}{\tau - \tau_0} \left(-\frac{5t^8}{2\tau_0^7} + \frac{10t^7}{\tau_0^6} - \frac{14t^6}{2\tau_0^5} + \frac{7t^5}{2\tau_0^4} \right), \\ c_{II}(t) &= \frac{1}{\tau - \tau_0} \left(t - \frac{\tau_0}{2} \right), \\ c_{III}(t) &= 1 + \frac{1}{\tau - \tau_0} \left(-\frac{5(\tau - t)^8}{2\tau_0^7} + \frac{10(\tau - t)^7}{\tau_0^6} - \frac{14(\tau - t)^6}{2\tau_0^5} + \frac{7(\tau - t)^5}{2\tau_0^4} \right). \end{aligned}$$

Using the reduced crane coordinates, the initial configuration of the rotary crane at $t_0 = 0$ is defined by $\bar{\boldsymbol{p}}_0 = [0 \ 5 \ 5]^T$, while the initial load coordinates are given by $\boldsymbol{x}_0 = [5 \ 0 \ -5]^T$.

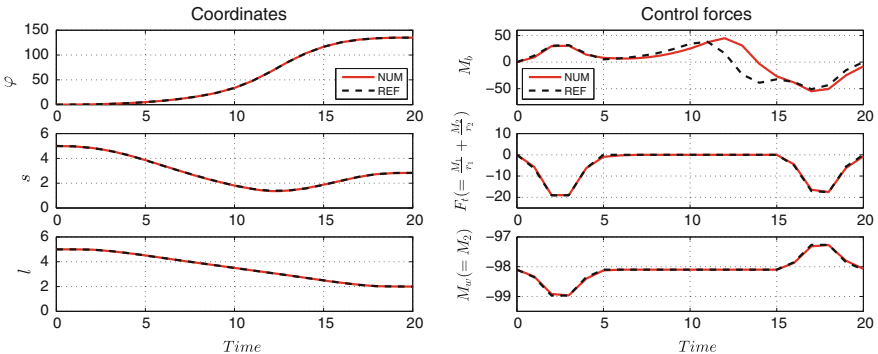


Fig. 1.3 Rotary crane: Comparison between the numerical results (NUM) obtained with $\Delta t = 1$ and the reference solution (REF) for the reduced crane coordinates $\bar{\boldsymbol{p}}(t)$

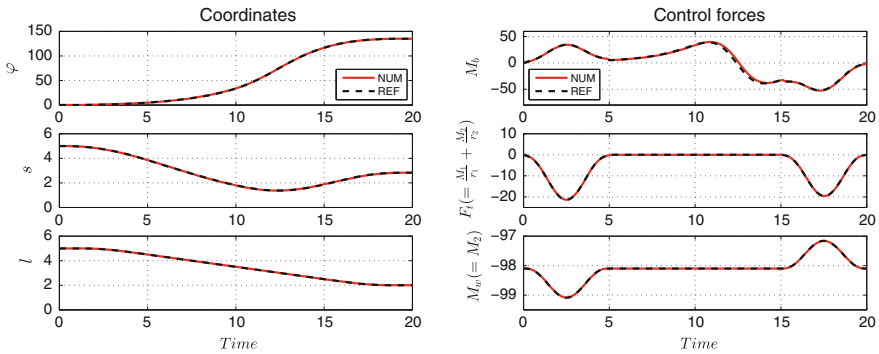


Fig. 1.4 Rotary crane: Comparison between the numerical results (NUM) obtained with $\Delta t = 0.1$ and the reference solution (REF) for the reduced crane coordinates $\bar{p}(t)$

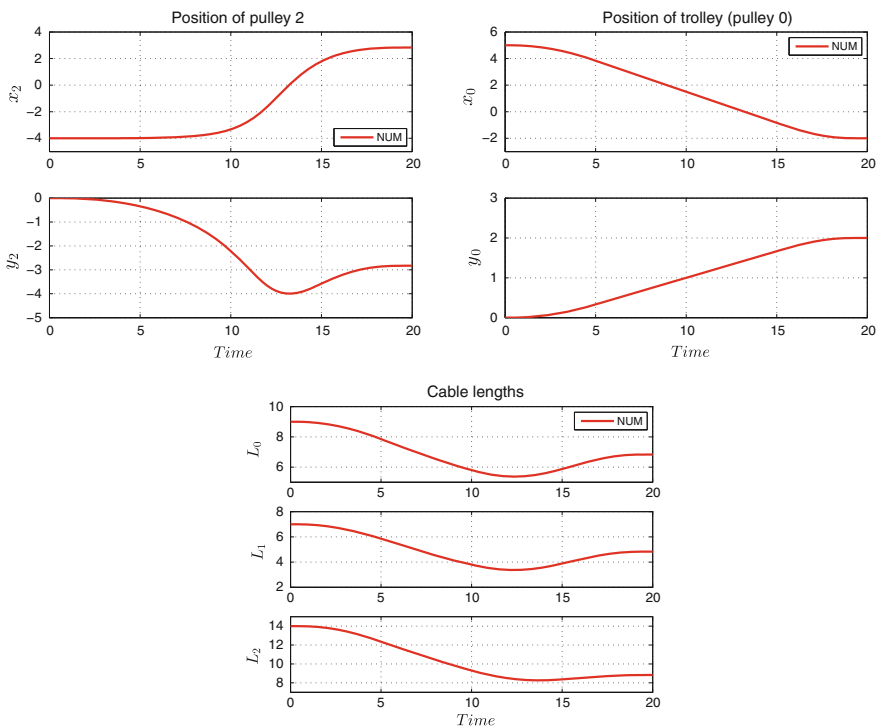


Fig. 1.5 Rotary crane: Numerical results (NUM) for the extended crane coordinates $p(t)$ obtained with $\Delta t = 0.1$

In Figs. 1.3 and 1.4, the numerical solution (NUM) is compared to the analytical reference solution (REF) obtained in [9]. It can be observed that the numerical solution converges to the reference solution when the time step size is reduced. In

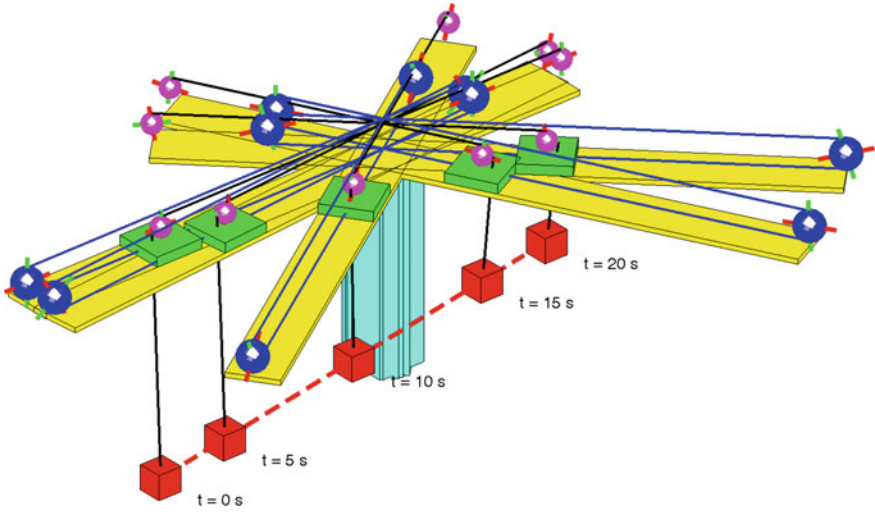


Fig. 1.6 Rotary crane (formulation in terms of the extended crane coordinates \mathbf{p}): Snapshots at specific points in time

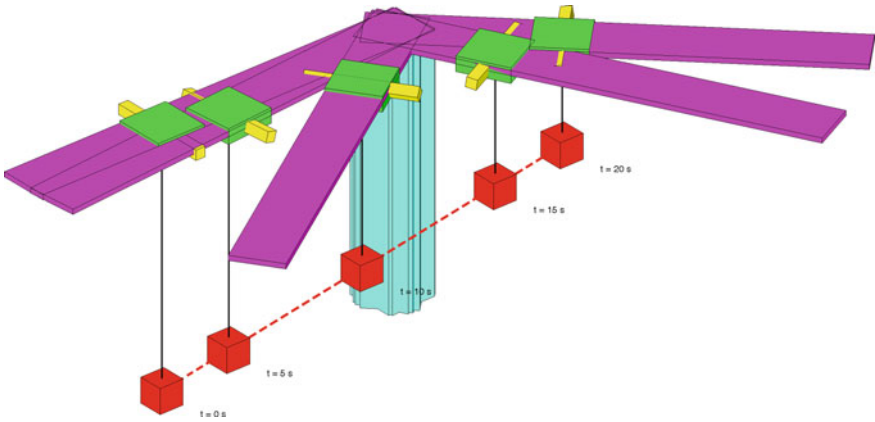


Fig. 1.7 Rotary crane (formulation in terms of the reduced crane coordinates $\bar{\mathbf{p}}$): Snapshots at specific points in time

in addition to that Fig. 1.5 displays the numerical solution for the extended crane coordinates \mathbf{p} . The two alternative formulations in terms of redundant coordinates (\mathbf{p} and $\bar{\mathbf{p}}$, respectively) yield practically indistinguishable results. The simulated motion of the rotary crane in terms of extended crane coordinates \mathbf{p} is illustrated in Fig. 1.6 with some snapshots at consecutive points in time. Similarly, snapshots obtained with the formulation in terms of the reduced crane coordinates $\bar{\mathbf{p}}$ are shown in Fig. 1.7.

1.6 Conclusions

We have dealt with a general formulation of underactuated mechanical systems subjected to both holonomic and servo constraints. It has been shown that index reduction by minimal extension makes possible the numerically stable inverse dynamics simulation of such systems. In contrast to our previous work [9] the present formulation does not restrict the level of redundancy of the coordinates used to model a specific mechanical system. Accordingly, truly rotationless formulations of multibody systems such as natural coordinates are now encompassed by our index reduction approach. The functional efficiency of the newly proposed method has been demonstrated in the framework of a general formulation of cranes. The corresponding rotationless formulation is characterized by constant inertia parameters and holonomic constraints that are at most quadratic in the coordinates. In forward dynamics, this type of formulation makes possible the design of structure-preserving integrators. Structure-preserving integrators for the inverse dynamics simulation of underactuated mechanical systems are not available yet. The present formulation is deemed to provide a good foundation for the development of such integrators which will be the goal of future work.

Acknowledgments The second author was supported by the ERC Advanced Grant ‘Modeling, Simulation and Control of Multi-Physics Systems’ MODSIMCONMP. The third author was supported by the Deutsche Forschungsgemeinschaft (DFG) under Grant BE 2285/12-1. This support is gratefully acknowledged.

References

1. Kirgetov VI (1967) The motion of controlled mechanical systems with prescribed constraints (servoconstraints). *J Appl Math Mech* 31(3):465–477
2. Blajer W (1997) Dynamics and control of mechanical systems in partly specified motion. *J Franklin Inst* 334B(3):407–426
3. Lam SH (1998) On Lagrangian dynamics and its control formulations. *Appl Math Comput* 91:259–284
4. Uhlar S, Betsch P (2009) A rotationless formulation of multibody dynamics: Modeling of screw joints and incorporation of control constraints. *Multibody Syst Dyn* 22(1):69–95
5. Blajer W, Seifried R, Kołodziejczyk K (2013) Diversity of servo-constraint problems for underactuated mechanical systems: a case study illustration. *Solid State Phenom* 198:473–482
6. Seifried R, Blajer W (2013) Analysis of servo-constraint problems for underactuated multibody systems. *Mech Sci* 4(1):113–129
7. Blajer W (2014) The use of servo-constraints in the inverse dynamics analysis of underactuated multibody systems. *J Comput Nonlinear Dynam* 9(4):041008/1-11
8. Kunkel P, Mehrmann V (2004) Index reduction for differential-algebraic equations by minimal extension. *Z Angew Math Mech (ZAMM)* 84(9):579–597
9. Altmann R, Betsch P, Yang Y (2015) Index reduction by minimal extension for the inverse dynamics simulation of cranes. Accepted for Publication in *Multibody System Dynamics*. doi:[10.1007/s11044-015-9471-x](https://doi.org/10.1007/s11044-015-9471-x)
10. Blajer W, Kołodziejczyk K (2004) A geometric approach to solving problems of control constraints: theory and a DAE framework. *Multibody Syst Dynam* 11(4):343–364

11. Kiss B, Lévine J, Müllhaupt P (1999) Modelling, flatness and simulation of a class of cranes. *Electr Eng* 43(3):215–225
12. Ascher UM, Petzold LR (1998) *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, Philadelphia
13. Blajer W, Kołodziejczyk K (2011) Improved DAE formulation for inverse dynamics simulation of cranes. *Multibody Syst Dynam* 25(2):131–143
14. Blajer W, Kołodziejczyk K (2005) A computational framework for control design of rotary cranes. In: Goicolea JM, Cuadrado J, Garcia Orden JC (ed) *Proceedings of ECCOMAS thematic conference on advances in computational multibody dynamics (CD-ROM)*, Madrid, June 21–24

Chapter 2

Enhancing the Performance of the DCA When Forming and Solving the Equations of Motion for Multibody Systems

Jeremy J. Laffin, Kurt S. Anderson and Mike Hans

Abstract This chapter provides an initial investigation into using the Graphics Processing Unit (GPU) (or similar hardware) to execute the Divide-and-Conquer Algorithm (DCA), which forms and solves the equations-of-motion for articulated multibody systems. The computational time required to form and solve the equations-of-motion of a simple n -length pendulum using the GPU is compared with a standard serial CPU implementation, a rudimentary parallelization on the CPU using OpenMP, and some combinations of the CPU and the GPU. The hybrid version uses the GPU for a select number of levels in the recursive sweeps and uses an OpenMP parallelization on a multi-core CPU for the remaining levels of recursion. The results demonstrate a significant performance increase when the GPU is used despite recursive algorithms being ill-suited to hardware designed for Single Instruction Multi-Data (SIMD). This is largely due to the tree-type structure of recursive processes, with half of the required operations being contained in the first level of recursion for a binary tree.

2.1 Introduction

Since computational performance is critically important for simulations to be used as an effective tool to study and design dynamic systems, the computing performance gains offered by GPUs should not be ignored. The GPU has been used to increase the computational performance of many tasks necessary to simulate multibody systems [6, 10, 11, 15, 16]. Since the GPU is designed to execute a very large number of simultaneous tasks (nominally SIMD), recursive algorithms in general, such as the DCA, are not well suited to be executed on GPU-type architecture. This

J.J. Laffin (✉) · K.S. Anderson
Rensselaer Polytechnic Institute, Troy, NY, USA
e-mail: laffij2@rpi.edu

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

M. Hans
Jet Propulsion Laboratory, Pasadena, CA, USA
e-mail: Michael.A.Hans.Jr@jpl.nasa.gov

is because each level of recursion is dependent on the previous level. Therefore, all tasks associated with the algorithm cannot be executed independently. The primary issue is the amount of data transfer that must occur when moving from one level of recursion to the next. However, the GPU can be leveraged to increase computational performance when using the DCA to form and solve the equations-of-motion for articulated multibody systems with a large number of degrees-of-freedom due to the inherent tree structure of DCA.

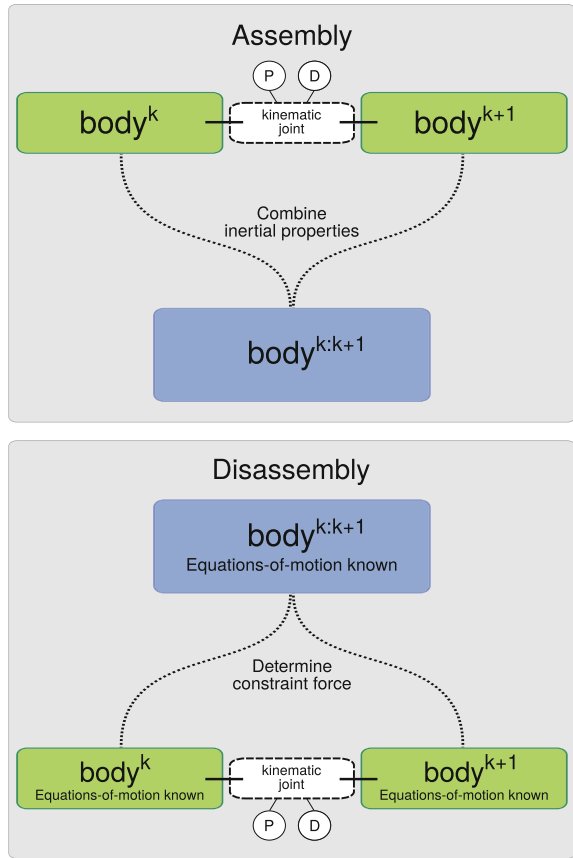
Computational performance of dynamic simulations is highly dependent on the nature of the underlying formulation and the number of generalized coordinates used to characterize the system. Therefore, algorithms that scale in a more desirable (lower order) fashion with the number of degrees-of-freedom are generally preferred when dealing with large ($N > 10$) systems. However, the utility of using simulations as a scientific tool is directly related to actual compute time. The DCA, and other top performing methods, have demonstrated the desirable property of the required compute time scaling linearly with ($O(n)$) with the number of degrees-of-freedom (n) and sublinearly ($O(\log n)$) when implemented in parallel. However for the DCA, total compute time could be further reduced using parallel hardware, such as the GPU, by exploiting the large number of independent operations involved in the first few levels of recursion.

2.1.1 The DCA for Forming and Solving Equations of Motion for Multibody Systems

The DCA was first introduced by Featherstone for both open-loop [5] and closed-loop [4] topologies and was notable for the level of coarse-grain parallelism it could achieve for unbranched systems. Additionally, there have been a number of modifications to the original method [1–3, 7–9, 13, 14, 17, 18]. The basic method is reproduced herein using the notation of Mukherjee and Anderson [12].

The DCA consists of two recursive processes: assembly and disassembly. These recursive processes take place using a hierarchical tree structure. The tree structure is defined by the kinematic joints connecting the bodies of the system. Typically the inboard and outboard joint of the body coincides with a reference point called a handle. The inverse inertial properties of two adjoining parent bodies are combined to represent a fictitious assembly (child body), see Fig. 2.1. This is possible because the kinematics of the joint are known, which allows the constraint forces acting at the connecting joint to be excluded from the equations-of-motion of the outboard handles of the parent bodies (the handles of the child body). The relative motion between the parent bodies is captured by an equation that describes the amount of motion happening in the directions of motion that are allowed by the joint, which are known. This process is repeated until there is only one body, the root body (see Fig. 2.2). At this point the boundary conditions are known and the equations-of-motion can be solved, which do not contain any of the constraint forces acting at the non-terminal joints.

Fig. 2.1 DCA kernel operations: These kernels both require that the motion allowed by the kinematic joint is mapped into a matrix containing the unit vectors of allowed motion (P), and a matrix containing the unit vectors of restricted motion (D)



In general, a body of the system may possess any number of handles, though the basic aspects of the method are most easily conveyed by discussing a chain system. For such systems, each body possesses two handles that connect the body to its inboard and outboard neighbors. The basic method involves writing the spatial equations-of-motion corresponding to these two reference points (handles) H_i^k $i = 1, 2$ on each body, as

$$A_1^k = \zeta_{11}^k F_{1c}^k + \zeta_{12}^k F_{2c}^k + \zeta_{13}^k, \quad (2.1)$$

$$A_2^k = \zeta_{21}^k F_{1c}^k + \zeta_{22}^k F_{2c}^k + \zeta_{23}^k, \quad (2.2)$$

and

$$A_1^{k+1} = \zeta_{11}^{k+1} F_{1c}^{k+1} + \zeta_{12}^{k+1} F_{2c}^{k+1} + \zeta_{13}^{k+1}, \quad (2.3)$$

$$A_2^{k+1} = \zeta_{21}^{k+1} F_{1c}^{k+1} + \zeta_{22}^{k+1} F_{2c}^{k+1} + \zeta_{23}^{k+1}. \quad (2.4)$$

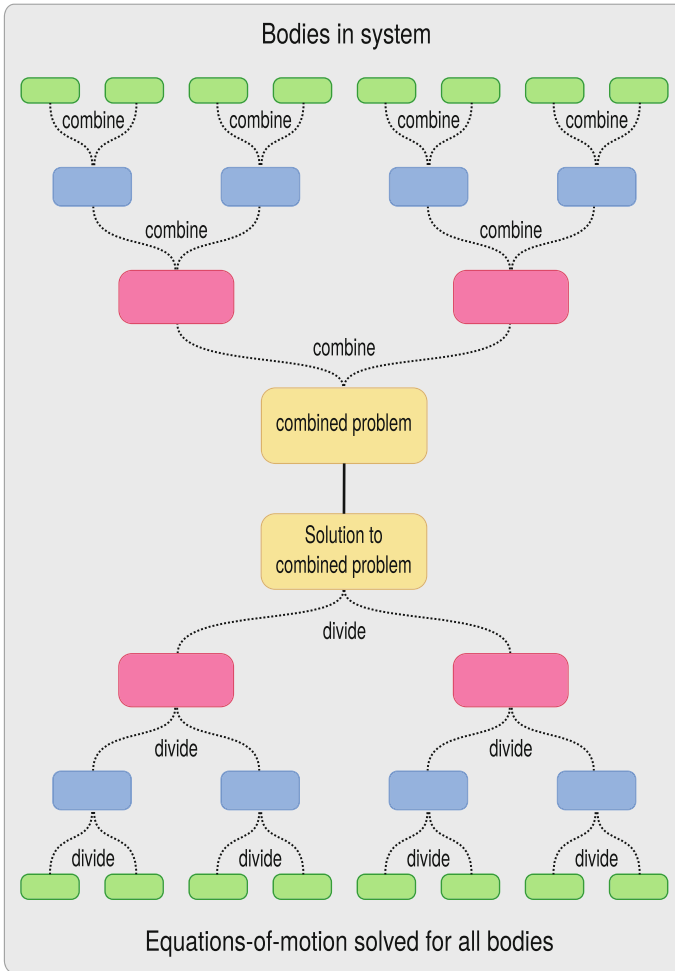


Fig. 2.2 Divide-and-Conquer Algorithm

These bodies, $Body^k$ and $Body^{k+1}$, connected by a kinematic joint j and therefore are subject to the kinematic constraint

$$P^j \dot{u} = A_1^{k+1} - A_2^k - \dot{P}^j u. \tag{2.5}$$

Here, the P matrix is a property of the kinematic joint whose columns define the unit vectors that are aligned with the directions of motion permitted by the kinematic joint. Similarly, the D matrix is that whose columns define the directions of motion

that are restricted by the joint. A_i^k , and F_i^k , are the spatial acceleration of, and force on, handle i respectively and are defined as

$$A_i^k = \begin{bmatrix} \boldsymbol{\alpha}^k \\ \mathbf{a}_i^k \end{bmatrix}, \quad (2.6)$$

and

$$F_{ic}^k = \begin{bmatrix} \boldsymbol{\tau}_i^k \\ \mathbf{f}_i^k \end{bmatrix}. \quad (2.7)$$

The rotational acceleration of $Body^k$ is $\boldsymbol{\alpha}^k$, \mathbf{a}_i^k is the translational acceleration of the reference point H_i^k , and $\boldsymbol{\tau}_i^k$ and \mathbf{f}_i^k are the constraint torques and forces acting at H_i^k , respectively. The ζ_{ij}^k ($i, j = 1, 2$) terms are the spatial matrix representations of the inverse inertial properties at the handles, while ζ_{i3}^k ($i = 1, 2$) contains applied forces acting on the body and other velocity dependent terms. The resulting set of equations, Eqs. (2.1)–(2.4), can be reduced by exploiting the fact that the constraint forces are equal and opposite, i.e., $F_{2c}^k = -F_{1c}^{k+1}$, and that the kinematics of the connecting joint are specified. Specifically, Eq. (2.5) describes the relative acceleration between connecting bodies using the generalized acceleration \dot{u} along known directions defined by the connecting joint partial velocity (mode of motion) P^j . The equations-of-motion for the assembled fictitious pseudo-body $Body^{k:k+1}$, at H_1^k and H_2^{k+1} can be expressed as

$$A_1^k = \zeta_{11}^{k:k+1} F_{1c}^k + \zeta_{12}^{k:k+1} F_{2c}^{k+1} + \zeta_{13}^{k:k+1} \quad (2.8)$$

and

$$A_2^{k+1} = \zeta_{21}^{k:k+1} F_{1c}^k + \zeta_{22}^{k:k+1} F_{2c}^{k+1} + \zeta_{23}^{k:k+1} \quad (2.9)$$

by algebraically eliminating the constraint forces at the connecting joint. The resulting Eqs. (2.8) and (2.9) are of the same form as the equations-of-motion for the handles of any generic body.

In the above equations, $\zeta_{ij}^{k:k+1}$ represents the inertial quantities of the fictitious pseudo-body resulting from the assembly of $Body^k$ and $Body^{k+1}$. For the derivation of the inverse inertial terms and the details of the assembly process, the reader is referred to the work of Featherstone [4] or Mukherjee and Anderson [12]. This assembly process is then repeated recursively, until only a single assembled pseudo-body remains (root body), as shown in Fig. 2.2. This is possible because the form of the equations-of-motion for the handles of an assembled body is indistinguishable from the form of the equations-of-motion for the handles of a generic body. The

assembly process yields the equations-of-motion associated with the two boundary handles

$$A_1^1 = \zeta_{11}^{1:n} F_{1c}^{1:n} + \zeta_{12}^{1:n} F_{2c}^{1:n} + \zeta_{13}^{1:n} \quad (2.10)$$

and

$$A_2^n = \zeta_{21}^{1:n} F_{1c}^{1:n} + \zeta_{22}^{1:n} F_{2c}^{1:n} + \zeta_{23}^{1:n}, \quad (2.11)$$

which are written in terms of only the spatial inertial quantities of all bodies in the system and the constraint forces acting at the two handles of the root body (boundary handles).

The spatial accelerations of, and constraint forces acting at H_1^1 and H_2^n can now be determined using the known boundary conditions. After determining these quantities, the disassembly process begins, in which all unknown spatial accelerations of the handles and constraint forces acting at all connecting joints are determined. This recursive process determines the constraint forces acting at a joint in terms of the constraint forces acting at the handles of the assembly, and the inertial properties of the assembled body, as

$$F_{1c}^{k+1} = W \zeta_{21}^k F_{1c}^k - W \zeta_{12}^{k+1} F_{2c}^{k+1} + Y. \quad (2.12)$$

The terms W and Y are terms containing inertial properties from the assembly of the two bodies, see Featherstone [4], or Mukherjee and Anderson [12] for derivation of these terms. Once this constraint force acting at a joint is determined, the spatial accelerations of the handles that are connected by this joint can be determined using Eqs. (2.8) and (2.9). This allows the computation of the generalized acceleration (\dot{u}) at the joint using Eq. (2.5).

2.1.2 Potential Challenges Executing the DCA on the GPU

Because this algorithm is by nature a recursive one, at first it seems ill-suited to run on the GPU architecture. This is because, by definition recursive algorithms can not be completely parallelized and require a minimum number of sequential operations. These sequential processes become problematic due to the potentially high cost of data transfer from the CPU memory to the GPU device memory. The lower connection speed is due to the physical arrangement of the CPU and the GPU and the type of connections used, which is continually improving.

The recursion becomes problematic, for the DCA, because the number of floating-point operations that need to be performed is low at the levels of recursion approaching the root body. Therefore, there is potentially a large amount of computational overhead in comparison to the number of floating-point operations that must be performed. This is contrary to the purpose for which GPUs were designed, which is

to produce a large amount of throughput. For this reason, it may seem that other algorithms which may scale more poorly with the number of bodies in the system, but can be more completely parallelized may achieve a lower compute time than the DCA or other recursive algorithms.

Although data transfer time is a significant source of overhead, there are others that are problematic for recursive algorithms, for example, kernel launch times and synchronization. Kernel launch time is particularly problematic for the DCA because the DCA consists of two operations, assembly and disassembly, which are each separate kernels and must be launched recursively. The recursive nature of the assembly and disassembly processes also necessitates that the kernels are synchronized at each level of recursion adding even more computational overhead.

2.1.3 Using the GPU Effectively With the DCA

Despite the apparent drawbacks of executing the DCA on the GPU, there are ways the inexpensive and powerful computing resources offered by the GPU can be utilized with the DCA. For large systems, there are a large number of independent assembly and disassembly operations that occur in the levels near the leaf level of the recursive tree. For a system of 2048 bodies the approximate operations counts for each level of recursion are given in Table 2.1. By examining the number of kernel operations (assembly and disassembly) necessary at each level of recursion, it is estimated that 50 % of the operations per kernel sweep are performed in the first level and 75 % are performed in the first two levels of recursion. By performing only three of the twelve levels of recursion on the GPU, 94 % of all the operations are parallelized. However, performing only one level's kernel operations on the GPU results in 50 % of the operations per solve can be parallelized with the penalty of associated with only one kernel launch and no synchronization penalty. Furthermore, 75 % of the operations can be parallelized incurring the data transfer and synchronization penalties of only one level of recursion in addition to the leaf level operations. As the system's size increases, using the GPU to perform the operations in these levels could allow much larger problems to be investigated by effectively halving or quartering the original problem seen by any CPU parallelization strategy.

By using the GPU in only the levels that have a relatively high number of parallel operations, the computational overhead compared to the number of floating-point operations is kept low. The CPU can then be used only for the remaining levels in

Table 2.1 Estimation of operations per level of recursion of the DCA

Level	Leaf	Leaf+1	Leaf+2	Leaf+3
Kernel operations	1024	1536	1792	1920
% Operations/Solve	50	75	88	94

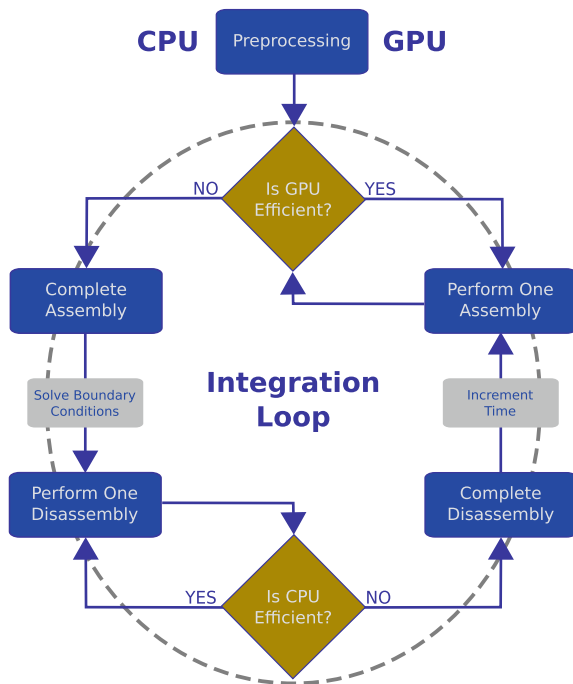
which there is a large number of sequential computations in comparison to a relatively low number of floating-point operations per level. In this way the GPU can be used to provide a large amount of throughput outweighing the overall overhead required.

Additionally, there are many ways that the data communication cost can be effectively hidden and the impact on overall simulation time can be minimized. For just one example of this consider a simulation of a bio-molecular system where coarse-graining has resulted in an articulated multibody system. In such an example, the data corresponding to the leaf level bodies can be transferred while computing the electrostatic forces.

Since the impact of the data communication cost associated with approximately 50 % of the operations can be mitigated, the penalty of using the GPU for two levels of recursion can be reduced to the cost of the data transfer between levels one and two ($\sim 1/4$ of the overall data), the synchronization between the two levels, and the second launch of the associated kernels.

Using the specifications of the known hardware it is possible to estimate whether or not using the GPU is beneficial. The flow-chart, shown in Fig. 2.3, demonstrates how this is performed for the DCA written as a recursive function. At each level of assembly an estimation is made whether the GPU or CPU is more efficient. This is repeated until all levels of assembly are completed, then the same estimation is made concerning the disassembly and each level of recursion is disassembled.

Fig. 2.3 Using DCA on the GPU: This figure describes how the DCA, written as a recursive function, can be effectively implemented in a way that uses the GPU only when advantageous



2.2 Implementation and Numerical Example

A simple numerical example was performed to investigate the computational advantages of executing the DCA on GPU architecture. The equations of motion are formed and solved using the DCA in a variety of parallel arrangements. The DCA is implemented in serial and parallel on multi-core CPU using OpenMP to parallelize the assembly and disassembly operations happening at each level. Other lower level CPU parallelization implementations are possible and with more effort may yield faster results. However, as a platform to investigate the scalability of the algorithm on the GPU as compared to a parallel CPU implementation, OpenMP is sufficient. The CPU was a Intel[®] Xeon[®] W3565 3.20 GHz having four cores.

The number of levels of recursive assembly, and subsequent disassembly was selected a priori and arbitrarily. However, with some simple functions to measure the data transfer time and kernel launch time the optimal number of levels for a particular hardware combination could be approximately determined. This was not performed because the information is specific to the hardware arrangement and is of little interest to other users. The levels of the DCA that are performed using the GPU were executed on a NVIDIA[®] Corporation GF108GL Quadro[®] 600. Therefore, the NVIDIA[®] CUDA[®] platform was used for the parallel implementation of the assembly and disassembly kernels.

2.2.1 Test Case

A multi-link pendulum, as shown in Fig. 2.4, was chosen as the test system to investigate how the computational time scales with the number of bodies using various approaches to GPU-parallelization. Although this example is 2D, all matrices and vectors are “full-sized” as they would be in the 3D case so that the data communication cost is nearly the same. The number of bodies was varied up to 2048. The number of levels of recursion that were executed on the GPU was varied from zero to all levels. For 2048 bodies this corresponds to eleven levels of recursive operations. The pendulum was released from rest at $q_i = 0$ for all i , and the simulation duration was a single time-step.

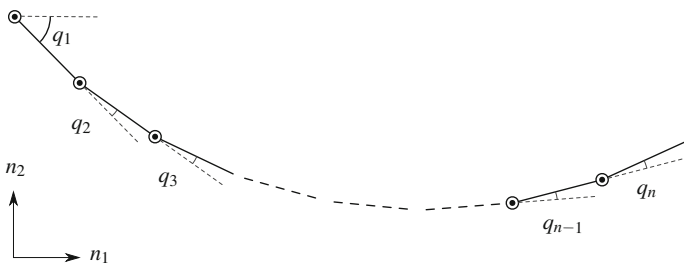


Fig. 2.4 Variable length pendulum test problem

Fig. 2.5 Full time-step: Serial run-time takes approximately 400 ms while the best hybrid solution takes approximately 240 ms. The OpenMP solution takes approximately 325 ms

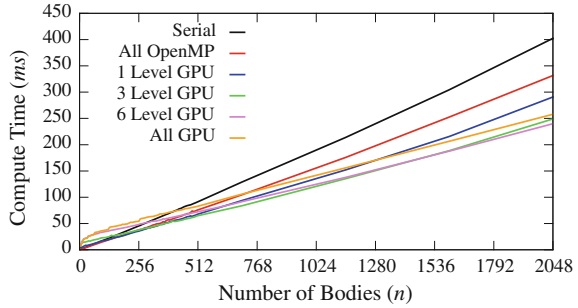
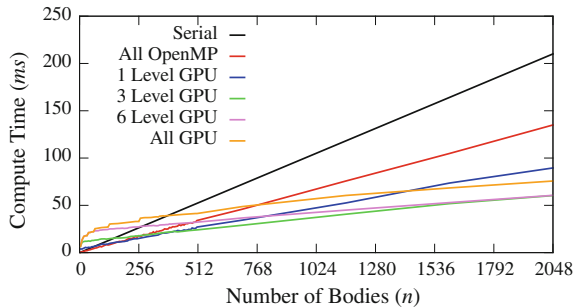


Fig. 2.6 DCA solve time: Serial run-time takes approximately 210 ms while the best hybrid solution takes approximately 60 ms. The OpenMP solution takes approximately 140 ms



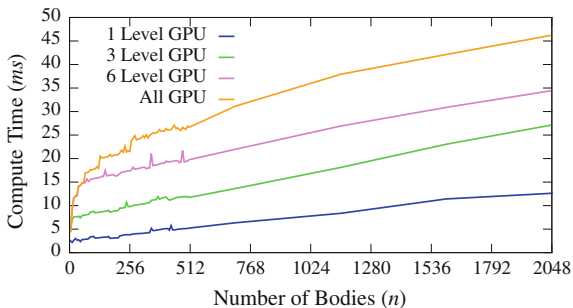
So that the effects of executing the DCA on GPU-type hardware can be better understood, the time to complete three different portions of the simulation are measured. These are the total time to complete one time-step of the simulation, the solve time of DCA, and the data transfer time. The total time to perform one time-step including all data transfer and operations needed to update the inverse inertial quantities for the next time step is shown in Fig. 2.5. The compute-time required to complete the assembly and disassembly sweeps, in which the equations of motion are formed and solved including determination of constraint forces acting at the joints, is plotted versus the number of bodies in the system in Fig. 2.6. Lastly, the time to transfer the necessary data to the device and then back to the host is shown in Fig. 2.7.

2.2.2 Results

Figure 2.5 demonstrates a decrease in computational time as compared to OpenMP when executing the DCA on GPU architecture, even for a moderately large number of bodies. Additionally, this figure also shows that parallelizing the assembly and disassembly operations of the leaf level, provides the largest reduction in compute time for this arrangement of 2048 bodies.

As the number of bodies increases, executing more levels of recursion provides still further reductions in computational time, although at a decreasing rate. It can

Fig. 2.7 Data transfer time: Data transfer time takes approximately 46 ms for all levels of recursion, which compares to 60 ms for solve time and 240 ms for the full time-step



also be seen from Fig. 2.5 that executing all levels of recursion outperforms the OpenMP solution near 512 bodies, which would mostly likely be reduced with a more sophisticated NVIDIA® CUDA® implementation of the DCA assembly and disassembly kernels. Conversely, this number may be increased with a more advanced CPU parallelization implementation. Finally, it can be seen that the speed-up gained from parallelizing all levels on the GPU is approximately double to that of the OpenMP parallelization for 2048 bodies.

By comparing the DCA solve-time, shown in Fig. 2.6, with the total compute-time of one time step, it can be seen that the DCA is approximately half of the total time associated with each time-step. This could most likely be lowered with a more thoughtful implementation of overhead operations required. Therefore, significant reductions of the solve-time directly and meaningfully impact the overall simulation time, which will only be more notable if other overhead is reduced. The solve time for parallelizing all levels of recursion on the GPU is approximately half of the solve-time compared to the OpenMP parallelization.

Additionally, the time required to transfer the data to and from the device, shown in Fig. 2.7, is a significant portion of the solve time. When performing all levels of recursion on the GPU, it can be seen that the solve-time is nearly entirely composed of data transfer time. It can also be inferred from this figure that most of data transfer cost is associated with initiating the transfer since the increase in compute-time related to more bodies (data) is much less than the increase due to more data transfers.

2.3 Conclusions

Despite the obvious drawbacks to executing a recursive algorithm on GPU architecture, a significant reduction in compute-time was observed when doing so with the DCA. For this implementation, the solve-time is reduced in half and the total compute-time is reduced by approximately a quarter when comparing the OpenMP and the best hybrid solution. This reduction in compute-time is due, for the most part, to the large amount of independent operations that occur in the first levels of recursion. Although a hybrid implementation using parallel processes on both the

CPU and the GPU provide the lowest compute times for moderately sized systems, the penalty for using the GPU exclusively becomes less important relatively quickly. Therefore for very large systems, such as those encountered in biomolecular simulations where the number of degrees-of-freedom (n) easily exceeds 10^5 , a hybrid approach may not be worthwhile.

Although the total time reduction to simulate a time-step is not as large as the solve-time reduction, it is still significant. The 25 % reduction in total-time demonstrates that implementing the DCA on GPU-type hardware is a viable option to achieve significant reduction in compute-time at low-cost. It also suggests that the desirable solve-time scaling properties of the DCA (and other recursive methods) may be preserved while significantly reducing the actual solve time using relatively inexpensive resources.

Furthermore, with the new features available with state-of-the-art compute-capability, such as dynamic parallelism, the penalty associated with using the GPU can be dramatically reduced, if not eliminated. This is because with this new technology, threads can launch kernels and therefore the data does not need to travel back to the host between levels of recursion. Additionally, threads can be launched at varying degrees of resolution. This not only allows the obvious benefit of not having to communicate back to the host device, but also allows the parallelism to be performed in a more straightforward manner. For example, consider a topology of many branches including various closed loops and other complex structures. Each branch can be given to a thread which recursively launches other threads for each nested structure until the final level of threads launches the ideal number of kernels to parallelize the decomposed topology. This technology changes the way in which recursive algorithms are thought to perform on the GPU.

References

1. 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
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. 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
4. Featherstone R (1999) A Divide-and-Conquer Articulated-body algorithm for parallel $O(\log(n))$ calculation of rigid-body dynamics. Part 1: l. *Int J Robot Res* 18(9):867–875
5. 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 Robot Res* 18(9):876–892
6. Khude N, Stanciulescu I, Melanz D, Negrut D (2013) Efficient parallel simulation of large flexible body systems with multiple contacts. *J Comput Nonlinear Dyn* 8(4):041003
7. Malczyk P, Frczek 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

8. Malczyk P, Mukherjee RM (2013) Parallel algorithm for modeling multi-rigid body system dynamics with nonholonomic constraints. In: Proceedings of the ASME 2013 international design engineering technical conferences, pp 1–9
9. Malczyk P, Frczek J, Cuadrado J (2010) Parallel index-3 formulation for real-time multibody dynamics simulations. In: Proceedings of the 1st joint international conference on multibody system dynamics. Lappeenranta, Finland
10. Mazhar H, Heyn T, Negrut D (2011) A scalable parallel method for large collision detection problems. *Multibody Syst Dyn* 26(1):37–55
11. Melanz D, Khude N, Jayakumar P, Negrut D (2013) A matrix-free NewtonKrylov parallel implicit implementation of the absolute nodal coordinate formulation. *J Comput Nonlinear Dyn* 9(1):011006
12. Mukherjee RM, Anderson KS (2006) Orthogonal complement based Divide-and-Conquer algorithm for constrained multibody systems. *Nonlinear Dyn* 48(1–2):199–215
13. 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
14. Mukherjee RM, Bhalerao KD, Anderson KS (2007) A divide-and-conquer direct differentiation approach for multibody system sensitivity analysis. *Struct Multidiscip Optim* 35(5):413–429
15. Negrut D, Tasora A, Mazhar H, Heyn T, Hahn P (2012) Leveraging parallel computing in multibody dynamics. *Multibody Syst Dyn* 27(1):95–117
16. Negrut D, Serban R, Mazhar H, Heyn T (2014) Parallel Computing in multibody system dynamics: why, when, and how. *J Comput Nonlinear Dyn* 9(4):041007
17. Poursina M, Anderson KS (2013) An extended Divide-and-Conquer algorithm for a generalized class of multibody constraints. *Multibody Syst Dyn* 29(3):235–254
18. Poursina M, Anderson KS (2013) Canonical ensemble simulation of biopolymers using a coarse-grained articulated generalized divide-and-conquer scheme. *Comput Phys Commun* 184(3):652–660. doi: 10.1016/j.cpc.2012.10.029, <http://www.sciencedirect.com/science/article/pii/S0010465512003724>

Chapter 3

Three-Dimensional Non-linear Shell Theory for Flexible Multibody Dynamics

Shilei Han and Olivier A. Bauchau

Abstract In flexible multibody systems, many components are approximated as shells. Classical theories, such as Kirchhoff and Reissner–Mindlin shell theories, are based on a priori kinematic assumptions. While such approach captures the kinetic energy of the system accurately, it cannot represent the strain energy adequately. Indeed, three-dimensional elasticity theory indicates that the normal material line will warp under load, leading to three-dimensional deformations and complex stress states. To overcome this problem, a novel three-dimensional shell theory is proposed in this paper. Kinematically, the problem is decomposed into a large rigid-normal-material-line motion and a warping field. The strains associated with the rigid-normal-material-line motion and the warping field are assumed to remain small. Consequently, the governing equations of the problem fall into two categories: the global equations describing geometrically exact shells and the local equations describing local deformations. The geometrically exact shell equations are nonlinear, two-dimensional equations, whereas the linear, local equations provide the detailed distribution of three-dimensional stress and strain fields. A shell stiffness matrix is found that reflects the effects of warping due to material heterogeneity and curvature. Three-dimensional stress and strain fields are recovered from the two-dimensional shell solution. The proposed approach is valid for anisotropic shells with arbitrary through-the-thickness lay-up configuration undergoing large motion but small strain.

S. Han (✉)
University of Michigan-Shanghai Jiao Tong University Joint Institute,
Shanghai 200240, China
e-mail: shilei.han@outlook.com

O.A. Bauchau
Department of Aerospace Engineering, University of Maryland,
College Park, MD 20742, USA

3.1 Introduction

Shells are structural components for which one dimension is far smaller than the other two. The midplane of the shell lies along its two long dimensions and its normal extends along the shorter dimension. The shell's thickness and material properties are assumed to vary smoothly over its midplane surface. When the midplane surface is planar, the shell is called a plate.

Numerous structures can be approximated as shells. The long, slender wings of an aircraft can be analyzed, to a first approximation, as beams, but a more refined analysis will treat the upper and lower skins of the wing as thin shells supported by ribs and longerons or stiffeners. The same can be said about helicopter or wind turbine blades. Buckling of the face sheets of wind turbine rotor blades is an important problem that cannot be captured by beam models. This instability, however, will be captured by shell models.

Solid mechanics theories describing shells, more commonly referred to as “shell theories,” provide useful tools for the analysis of these commonly used structural components. Although more sophisticated formulations based on three-dimensional elasticity theory could be used to analyze shells, the associated computational burden is often too heavy. Shell theories reduce the analysis of complex, three-dimensional structures to two-dimensional problems. The main goal of the proposed theory is to approximate the three-dimensional shell-like structure with a two-dimensional model, while retaining an accurate representation of the local, three-dimensional stress and strain fields through the thickness of the shell.

Several shell theories have been developed based on various assumptions that lead to different levels of accuracy. One of the simplest theory is due to Kirchhoff who analyzed the bending of thin plates. Shell theories based on Kirchhoff assumptions [1–3] are used in civil, mechanical, and aerospace applications, although shear deformable theories, called “Reissner–Mindlin shell theories” [4–6], have also found wide acceptance.

In many applications, shells are complex built-up structures. In aeronautical constructions, for instance, the increased use of laminated composite materials leads to heterogeneous, anisotropic structures. Layers of anisotropic material are stacked through the thickness of the shell. This new type of structural component prompted the development of new plate and shell theories [7–9], often based on classical lamination theory [10, 11].

Further refinements followed with the goal of capturing the intricate three-dimensional stress field that develops under load, with special emphasis on interlaminar shear stresses. The various approaches fall into two categories: higher-order and layer-wise shell theories. Higher-order shell theories assume higher-order displacement fields for the entire normal material line [12–14]. In layer-wise theories [15, 16], the displacement field of the normal material line for each layer is independent of that of the others, with a simple C_0 continuity condition at the layer boundaries. Although these approaches lead to higher accuracy, the number of unknowns

increases considerably. A comprehensive review of the literature is given by Noor and Burton [17].

Efficient shell models can be obtained more rigorously from three-dimensional elasticity through dimensional reduction techniques that split the original problem into a two-dimensional analysis over the shell's midplane surface and a one-dimensional, through-the-thickness linear analysis. The shell's stiffness matrix is a by-product of the dimensional reduction process, which also enables the recovery of three-dimensional stress fields. These approaches, derived from three-dimensional elasticity theory directly, can handle laminated shells made of anisotropic composite materials without increasing the total number of unknowns.

Asymptotic and multiscale analysis methods have been the tools of choice for dimensional reduction. For shell problems, the thickness, denoted h , is typically far smaller than a representative span-wise dimension, denoted a , and hence, the aspect ratio, $\varepsilon = h/a$, is a small parameter. Consequently, the gradients of stress and displacement components over the shell's midplane surface are smaller than those through its thickness. Asymptotic and multiscale methods expand the solution in terms of the aspect ratio, leading to a rational decomposition of the three-dimensional problem into two-dimensional equations over the shell's midplane and a through-the-thickness problem. Based on this approach, Friedrichs and Dressler [18] and Kalamkarov [19] investigated isotropic and heterogenous plate problems, respectively.

In the Variational Asymptotic Method (VAM) proposed by Berdichevsky [20, 21], asymptotic analysis is applied to the energy functional. A unified theory based on VAM, presenting both linear, one-dimensional through-the-thickness analysis, and nonlinear, two-dimensional analysis over the shell's midplane surface was further refined by Sutyryn and Hodges [22, 23], and Yu et al. [24, 25]. The shell's 8×8 stiffness matrix relating the eight deformation measures to the corresponding eight stress resultants is a by-product of the linear, through-the-thickness analysis. Furthermore, the strain field through the shell's thickness can be recovered once the strain measures are known.

More recently, Kim [26] proposed a finite element based asymptotic analysis for generally anisotropic plate problems. The dimensional reduction process is based on a formal asymptotic expansion, which splits the three-dimensional plate problem into two sets of recursive equations: a set of one-dimensional local recursive problems and a set of two-dimensional global recursive problems.

It is not necessary to use asymptotic expansion methods to tackle plate analysis. Giavotto et al. [27] used a finite element based, semi-discretization approach to solve beam problems presenting arbitrary cross-sectional geometries and material properties. An adaptation of this approach to anisotropic, heterogeneous laminated composite plates was proposed by Masarati and Ghiringhelli [28], who found solutions of the three-dimensional equilibrium equations through the plate's thickness. These solutions then yield the plate's compliance matrix and local stress fields can be recovered.

The Representative Volume Element (RVE) approach is a common tool for multiscale analysis. Recently, Gruttmann and Wagner [29] used a through-the-thickness

RVE to develop local/global plate and shell models. The displacement field of the RVE is split into rigid normal material line and warping components. The local RVE and global plate models are coupled at the lateral surfaces of the RVE. The coupled governing equations of the two-scale model are solved simultaneously.

In a recent paper [30] by the authors, a dimensional reduction technique for plate problems was proposed. A semi-discretization of the general equations of three-dimensional elasticity was performed, defining the local model. The equations relating the stress resultants, the plate's strain measures, and the warping field of the normal material line were derived from a linear combination of the equations of the local model, defining the global model. Based on a set of power series solutions, the local problem for plate-like structures was reduced to the corresponding global problem and local stress and strain fields can be recovered from the global solution.

In this paper, the approach is extended to shells undergoing large displacements and rotations. Kinematically, the problem is decomposed into a large rigid-normal-material-line motion and a warping field. The strain measures associated with the rigid-normal-material-line motion and the warping field are assumed to remain small. Consequently, the governing equations of the problem fall into two categories: the global equations describing geometrically exact shells and the local equations describing local deformations. The geometrically exact shell equations are nonlinear, two-dimensional equations, whereas the linear, local equations provide the detailed distribution of three-dimensional stress and strain fields. A shell stiffness matrix is found that reflects the effects of warping due to material heterogeneity and shell curvature. Three-dimensional stress and strain fields are recovered from the two-dimensional shell solution. The proposed approach is valid for anisotropic shells with arbitrary through-the-thickness lay-up configuration undergoing large motion but small strain.

The following assumptions are made: (1) the shell's first and second fundamental forms remain constant over its midplane surface; (2) the shell's thickness and material properties are uniform over its midplane surface; (3) the shell undergoes large motion but the strains and warping displacements remain small. Due to these three assumptions, the governing equations of the problem can be cast into a set of partial differential equations with constant coefficients. Although material properties remain uniform, heterogeneous, anisotropic materials can be accommodated. For many practical problems, assumptions (1) and (2) might not be verified exactly because the shell's configuration and physical properties vary over its midplane surface. For as long as such gradients remain small, the proposed approach should provide good accuracy.

The paper is organized as follows: the kinematics of the problem and the governing equations of the Saint-Venant's problem are presented in Sects. 3.2 and 3.3, respectively. The reduction of the three-dimensional governing equations to shell's equations and recovery of three-dimensional stress fields are the focus of Sects. 3.4 and 3.5; numerical examples are presented in the last section.

3.2 Kinematics of the Problem

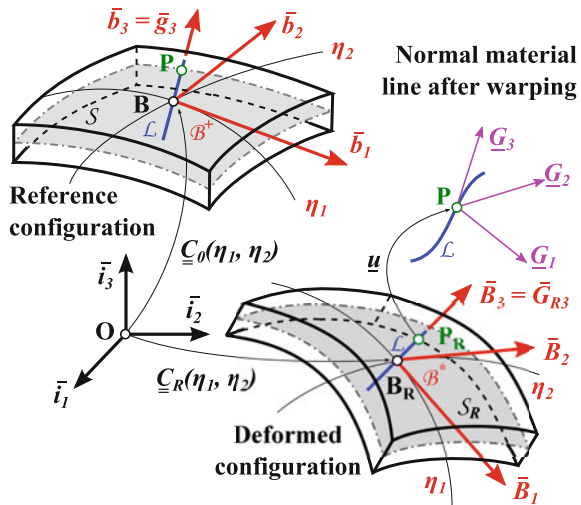
Figure 3.1 depicts a shell of thickness h , midplane surface S , and normal material line \mathcal{L} , in its reference and deformed configurations. The volume of the shell is generated by sliding line \mathcal{L} over the shell's midplane. A set of material coordinates is selected as follows: curvilinear coordinate η_1 and η_2 describe curves within surface S and η_3 is along line \mathcal{L} . Point \mathbf{B} is located at the intersection of the normal material line with the shell's midplane. Let $\underline{r}_B(\eta_1, \eta_2)$ denote the position vector of point \mathbf{B} with respect to the origin of the reference frame, $\mathcal{F}_I = [\mathbf{O}, \mathcal{I} = (\bar{i}_1, \bar{i}_2, \bar{i}_3)]$.

In the reference configuration, material frame $\mathcal{F}_B = [\mathbf{B}, \mathcal{B}^\dagger = (\bar{b}_1, \bar{b}_2, \bar{b}_3)]$ is defined, where unit vectors \bar{b}_1 and \bar{b}_2 are in the plane tangent to surface S at point \mathbf{B} and unit vector \bar{b}_3 is the normal to surface S at point \mathbf{B} . Unit vectors \bar{b}_α are defined as $a_\alpha \bar{b}_\alpha = \partial \underline{r}_B / \partial \eta_\alpha$, where Lamé's coefficients, $a_\alpha = \|\partial \underline{r}_B / \partial \eta_\alpha\|$, represent the length of the tangent vectors. Throughout this paper, Greek subscripts are assumed to take values of 1 to 2. For convenience, it is assumed that curvilinear coordinates η_1 and η_2 have been selected such that unit vectors \bar{b}_1 and \bar{b}_2 are mutually orthogonal. Normal material line \mathcal{L} is along unit vector $\bar{b}_3 = \bar{b}_1 \bar{b}_2$.

Basis \mathcal{B}^\dagger is a function of curvilinear variables η_α ; the rotation tensor that brings basis \mathcal{I} to basis \mathcal{B}^\dagger is denoted as $\underline{R}_0(\eta_1, \eta_2)$. The motion tensor [31] that brings the inertial frame \mathcal{F}_I to frame \mathcal{F}_B is

$$\underline{C}_0(\eta_1, \eta_2) = \begin{bmatrix} \underline{R}_0 & \tilde{r}_B \underline{R}_0 \\ \underline{0} & \underline{R}_0 \end{bmatrix}. \tag{3.1}$$

Fig. 3.1 Configuration of the shell in its reference and deformed configurations



The components of the shell's generalized curvature vector in its reference configuration are then $a_\alpha \tilde{k}_\alpha^\dagger = \underline{\underline{C}}_0^{-1} \underline{\underline{C}}_{=0,\alpha} = -\underline{\underline{C}}_{=0,\alpha}^{-1} \underline{\underline{C}}_0$, where notation $(\cdot)^\dagger$ indicates tensor components resolved in basis $\underline{\underline{B}}^\dagger$, and the second equality results from identity $\underline{\underline{C}}_0^{-1} \underline{\underline{C}}_0 = \underline{\underline{I}}$. The explicit expression of the curvature vector is

$$\tilde{\mathcal{K}}_\alpha^\dagger = \begin{bmatrix} \tilde{k}_\alpha^\dagger & \tilde{t}_\alpha^\dagger \\ \underline{\underline{0}} & \tilde{k}_\alpha^\dagger \end{bmatrix}. \quad (3.2)$$

The associated generalized curvature vector is $\underline{\underline{k}}_\alpha^{\dagger T} = \{\underline{\underline{t}}_\alpha^{\dagger T}, \underline{\underline{k}}_\alpha^{\dagger T}\}$. It is verified easily that $a_\alpha \underline{\underline{t}}_\alpha^\dagger = \underline{\underline{R}}_0^T \underline{\underline{r}}_{B,\alpha}$ and $a_\alpha \underline{\underline{k}}_\alpha^\dagger = \underline{\underline{R}}_0^T \underline{\underline{R}}_{0,\alpha}$. The components of the curvature vectors are $\underline{\underline{k}}_1^\dagger = \{-k_{12}^\dagger, k_{11}^\dagger, k_{13}^\dagger\}^T$ and $\underline{\underline{k}}_2^\dagger = \{-k_{22}^\dagger, k_{21}^\dagger, k_{23}^\dagger\}^T$, where $a_\alpha a_\beta k_{\alpha\beta}^\dagger = -b_{\alpha\beta}^\dagger$, $a_1 a_2 k_{13}^\dagger = -\partial a_1 / \partial \eta_2$, and $a_1 a_2 k_{23}^\dagger = \partial a_2 / \partial \eta_1$. The shell's first and second fundamental forms are assumed to remain constant over its midplane surface and hence, the partial derivatives of Lamé's coefficients and curvature components k_{13}^\dagger and k_{23}^\dagger vanish. The components of the second fundamental form of the midplane surface are denoted $b_{\alpha\beta}^\dagger$ and defined as

$$\bar{b}_3^{\dagger T} d\underline{\underline{r}}_B = b_{\alpha\beta}^\dagger d\eta_\alpha d\eta_\beta. \quad (3.3)$$

Clearly, $b_{\alpha\beta}^\dagger = \underline{\underline{b}}_3^{\dagger T} \partial^2 \underline{\underline{r}}_B / (\partial \eta_\alpha \partial \eta_\beta)$. For the remainder of this paper it is assumed that curvilinear coordinates η_1 and η_2 define the lines of curvatures, i.e., κ_{12}^\dagger and κ_{21}^\dagger vanish. The principal curvatures are then $k_{11}^\dagger = 1/\rho_1$ and $k_{22}^\dagger = 1/\rho_2$, where ρ_1 and ρ_2 are the principal radii of curvature.

3.2.1 The Reference Configuration

The position vector of an arbitrary material point \mathbf{P} of the shell in its reference configuration now becomes

$$\underline{\underline{r}}(\eta_1, \eta_2, \eta_3) = \underline{\underline{r}}_B(\eta_1, \eta_2) + \eta_3 \bar{b}_3. \quad (3.4)$$

The associated base vectors are then

$$\underline{\underline{g}}_1 = \partial \underline{\underline{r}}_B / \partial \eta_1 + \eta_3 \partial \bar{b}_3 / \partial \eta_1 = a_1 (1 + \eta_3 / \rho_1) \bar{b}_1, \quad (3.5a)$$

$$\underline{\underline{g}}_2 = \partial \underline{\underline{r}}_B / \partial \eta_2 + \eta_3 \partial \bar{b}_3 / \partial \eta_2 = a_2 (1 + \eta_3 / \rho_2) \bar{b}_2, \quad (3.5b)$$

$$\underline{\underline{g}}_3 = \bar{b}_3. \quad (3.5c)$$

The components of the base vectors resolved in basis $\underline{\underline{B}}^\dagger$ are $\underline{\underline{g}}_i^\dagger = \underline{\underline{R}}_0^T \underline{\underline{g}}_i$, $i = 1, 2, 3$; in the rest of this paper subscript i is assumed to take values of 1, 2, and 3. These results

are recast in a compact form

$$\underline{\underline{g}}^\dagger = [\underline{\underline{g}}_1^\dagger \ \underline{\underline{g}}_2^\dagger \ \underline{\underline{g}}_3^\dagger] = \begin{bmatrix} a_1 \chi_1 & 0 & 0 \\ 0 & a_2 \chi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.6)$$

where two non-dimensional parameters were introduced $\chi_1 = 1 + \eta_3/\rho_1$ and $\chi_2 = 1 + \eta_3/\rho_2$.

3.2.2 The Fictitious Rigid-Normal-Line Motion

Figure 3.1 shows the configuration of the shell in its deformed configuration. The displacement field of the material line is decomposed into two parts: a fictitious rigid-normal-material-line motion (abbreviated as “rigid-normal motion”) and an arbitrary warping field. The rigid-normal motion is characterized by five degrees of freedom only, three displacements and two rotations, because the rotation of the normal line about its own axis, called “drilling rotation,” is immaterial.

While the kinematics description does not require the drilling rotation, its introduction allows the definition of a convected material frame in which strain and stress tensors can be resolved conveniently. Furthermore, the shell becomes a Cosserat solid described by displacement and rotation fields. Due to the fictitious rigid-normal motion, material points \mathbf{B} and \mathbf{P} move to points \mathbf{B}_R and \mathbf{P}_R , respectively, see Fig. 3.1, and the convected material frame, $\mathcal{F}_R = [\mathbf{B}_R, \mathcal{B}^* = (\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \bar{\mathbf{B}}_3)]$, is defined. The drilling rotation, to be selected later, allows the unambiguous definition of basis \mathcal{B}^* and rotation tensor $\underline{\underline{R}}(\eta_1, \eta_2)$ brings basis \mathcal{B}^\dagger to basis \mathcal{B}^* . The shell’s midplane surface is now \mathcal{S}_R and let $\underline{\underline{r}}_{B_R}$ be the position vector of point \mathbf{B}_R with respect to point \mathbf{O} . The motion tensor that brings frame \mathcal{F}_I to \mathcal{F}_R is now

$$\underline{\underline{C}}_R(\eta_1, \eta_2) = \begin{bmatrix} (\underline{\underline{R}} \underline{\underline{R}}_0) & \tilde{r}_{B_R}(\underline{\underline{R}} \underline{\underline{R}}_0) \\ \underline{\underline{0}} & (\underline{\underline{R}} \underline{\underline{R}}_0) \end{bmatrix}. \quad (3.7)$$

The components of the shell’s generalized curvatures in its fictitious configuration are then $a_\alpha \tilde{\mathcal{K}}_\alpha^* = \underline{\underline{C}}_R^{-1} \underline{\underline{C}}_{R,\alpha}^*$, where notation $(\cdot)^*$ indicates tensor components resolved in basis \mathcal{B}^* . Clearly,

$$\tilde{\mathcal{K}}_\alpha^* = \begin{bmatrix} \tilde{K}_\alpha^* & \tilde{T}_\alpha^* \\ \underline{\underline{0}} & \tilde{K}_\alpha^* \end{bmatrix}, \quad (3.8)$$

where $a_\alpha \underline{\underline{T}}_\alpha^* = (\underline{\underline{R}} \underline{\underline{R}}_0)^T \underline{\underline{r}}_{B_R,\alpha}$ and $a_\alpha \tilde{K}_\alpha^* = (\underline{\underline{R}} \underline{\underline{R}}_0)^T (\underline{\underline{R}} \underline{\underline{R}}_0)_{,\alpha}$. The associated generalized curvature vector is $\underline{\underline{K}}_\alpha^{*T} = \{\underline{\underline{T}}_\alpha^{*T}, \underline{\underline{K}}_\alpha^{*T}\}$. The following notation is introduced: $T_{\alpha i}^*$ denotes the i th component of array $\underline{\underline{T}}_\alpha^{*T}$, while the components of the curvature vectors are $\underline{\underline{K}}_1^* = \{-K_{12}^*, K_{11}^*, K_{13}^*\}^T$ and $\underline{\underline{K}}_2^* = \{-K_{22}^*, K_{21}^*, K_{23}^*\}^T$.

The virtual motion vector, $\underline{\delta\mathcal{U}}$, is introduced by analogy with the generalized curvature vector as $\underline{\delta\tilde{\mathcal{U}}}^* = \underline{\mathcal{C}}_{\underline{R}}^{-1} \underline{\delta\mathcal{C}}_{\underline{R}}$. Taking a derivative of virtual motion vector yields $\underline{\delta\tilde{\mathcal{U}}}_{,\alpha}^* = \underline{\mathcal{C}}_{\underline{R},\alpha}^{-1} \underline{\delta\mathcal{C}}_{\underline{R}} + \underline{\mathcal{C}}_{\underline{R}}^{-1} \underline{\delta\mathcal{C}}_{\underline{R},\alpha}$ while taking a variation of generalized curvature vector leads to $a_\alpha \underline{\delta\tilde{\mathcal{K}}}_\alpha^* = \underline{\delta\mathcal{C}}_{\underline{R},\alpha}^{-1} \underline{\mathcal{C}}_{\underline{R},\alpha} + \underline{\mathcal{C}}_{\underline{R}}^{-1} \underline{\delta\mathcal{C}}_{\underline{R},\alpha}$. Subtracting these two results then yields $\underline{\delta\tilde{\mathcal{U}}}_{,\alpha}^*/a_\alpha - \underline{\delta\tilde{\mathcal{K}}}_\alpha^* = \underline{\delta\tilde{\mathcal{U}}}^* \tilde{\mathcal{K}}_\alpha^* - \tilde{\mathcal{K}}_\alpha^* \underline{\delta\tilde{\mathcal{U}}}^*$. When expressed in the vector form, this leads to

$$\underline{\delta\tilde{\mathcal{U}}}_{,\alpha}^*/a_\alpha = \underline{\delta\tilde{\mathcal{K}}}_\alpha^* - \tilde{\mathcal{K}}_\alpha^* \underline{\delta\tilde{\mathcal{U}}}^*. \quad (3.9)$$

These equation defines the transpositional relationships that links the partial derivatives of the virtual motion and variation of generalized curvatures.

As proposed by Fox and Simo [32], the drilling rotation is defined by the following constraint, to be enforced via a Lagrange multiplier,

$$T_{12}^* - T_{21}^* = 0. \quad (3.10)$$

This constraint implies the equality of angles $\arccos(\underline{T}_1^T \bar{\mathbf{B}}_1)$ and $\arccos(\underline{T}_2^T \bar{\mathbf{B}}_2)$.

In the fictitious rigid-normal configuration, the position vector of material point \mathbf{P}_R becomes

$$\underline{r}_{P_R}(\eta_1, \eta_2, \eta_3) = \underline{r}_{B_R} + \eta_3 \bar{\mathbf{B}}_3. \quad (3.11)$$

The base vectors in the fictitious configuration, $\underline{\mathcal{G}}_{Ri} = \partial \underline{r}_{P_R} / \partial \eta_i$, now become

$$\underline{\mathcal{G}}_{R1} = \partial \underline{r}_{B_R} / \partial \eta_1 + \eta_3 \partial \bar{\mathbf{B}}_3 / \partial \eta_1 = a_1 [\underline{T}_1 + \eta_3 K_{11} \bar{\mathbf{B}}_1 + \eta_3 K_{12} \bar{\mathbf{B}}_2], \quad (3.12a)$$

$$\underline{\mathcal{G}}_{R2} = \partial \underline{r}_{B_R} / \partial \eta_2 + \eta_3 \partial \bar{\mathbf{B}}_3 / \partial \eta_2 = a_2 [\underline{T}_2 + \eta_3 K_{21} \bar{\mathbf{B}}_1 + \eta_3 K_{22} \bar{\mathbf{B}}_2], \quad (3.12b)$$

$$\underline{\mathcal{G}}_{R3} = \bar{\mathbf{B}}_3. \quad (3.12c)$$

The components of these base vectors resolved in basis \mathcal{B}^* are simply $\underline{\mathcal{G}}_{Ri}^* = (\underline{\mathcal{R}} \underline{\mathcal{R}}_0)^T \underline{\mathcal{G}}_{Ri}$, and can be recast in a compact form as

$$\underline{\mathcal{G}}_{Ri}^* = [\underline{\mathcal{G}}_{R1}^* \ \underline{\mathcal{G}}_{R2}^* \ \underline{\mathcal{G}}_{R3}^*] = [\underline{T}_1^* + \eta_3 \tilde{K}_1^* \bar{\mathbf{t}}_3 \ \underline{T}_2^* + \eta_3 \tilde{K}_2^* \bar{\mathbf{t}}_3 \ \bar{\mathbf{t}}_3]. \quad (3.13)$$

3.2.3 The Deformed Configuration

The displacement field of the material line has been decomposed into two parts: the rigid-normal motion and an arbitrary warping field, denoted $\underline{u}(\eta_1, \eta_2, \eta_3)$, which describes the displacement of material point \mathbf{P} with respect to point \mathbf{P}_R , see Fig. 3.1. Because the warping field is arbitrary, it also includes a rigid-body motion, and hence, rigid-body motions are double counted. This ambiguity of the formulation will be resolved later, based on physical arguments. In the deformed configuration,

the position vector of a material point becomes

$$\underline{r}_P(\eta_1, \eta_2, \eta_3) = \underline{r}_{P_R}(\eta_1, \eta_2, \eta_3) + (\underline{R} \underline{R}_0) \underline{u}^*. \quad (3.14)$$

The base vectors in the deformed configuration, $\underline{G}_i = \partial \underline{r}_P / \partial \eta_i$, now become

$$\underline{G}_1 = \underline{G}_{R1} + (\underline{R} \underline{R}_0)(a_1 \tilde{K}_1^* \underline{u}^* + \underline{u}_{,1}^*), \quad (3.15a)$$

$$\underline{G}_2 = \underline{G}_{R2} + (\underline{R} \underline{R}_0)(a_2 \tilde{K}_2^* \underline{u}^* + \underline{u}_{,2}^*), \quad (3.15b)$$

$$\underline{G}_3 = \underline{G}_{R3} + (\underline{R} \underline{R}_0) \underline{u}_{,3}^*. \quad (3.15c)$$

The components of these base vectors resolved in basis \mathcal{B}^* are $\underline{G}_i^* = (\underline{R} \underline{R}_0)^T \underline{G}_i$, and can be recast in a compact form

$$\underline{G}^* = [\underline{G}_1^* \ \underline{G}_2^* \ \underline{G}_3^*] = \underline{G}_R^* + \underline{G}_{W1}^* + \underline{G}_{W2}^*, \quad (3.16)$$

where matrices \underline{G}_{W1}^* and \underline{G}_{W2}^* represent the effects of warping

$$\underline{G}_{W1}^* = [a_1 \tilde{k}_1^* \underline{u}^* + \underline{u}_{,1}^* \ a_2 \tilde{k}_2^* \underline{u}^* + \underline{u}_{,2}^* \ \underline{u}_{,3}^*], \quad (3.17a)$$

$$\underline{G}_{W2}^* = [a_1 \tilde{\varkappa}_1^* \underline{u}^* \ a_2 \tilde{\varkappa}_2^* \underline{u}^* \ 0], \quad (3.17b)$$

and $\underline{\varkappa}_\alpha^* = \underline{K}_\alpha^* - \underline{k}_\alpha^\dagger$ are the elastic curvatures. It is assumed that strain measures and warping displacements are small and hence, second order terms such as $\tilde{\varkappa}_1^* \underline{u}^*$ and $\tilde{\varkappa}_2^* \underline{u}^*$ are negligible. The base vectors in the deformed configuration now reduce to

$$\underline{G}^* \approx \underline{G}_R^* + \underline{G}_{W1}^*. \quad (3.18)$$

3.2.4 Strain Components

The components of deformation gradient tensor in the mixed bases \mathcal{B}^* and \mathcal{B}^\dagger are

$$\underline{F}^{*\dagger} = \underline{G}^* \underline{g}^{\dagger-1} \approx \underline{G}_R^* \underline{g}^{\dagger-1} + \underline{G}_{W1}^* \underline{g}^{\dagger-1} = \underline{F}_R + \underline{E}_W, \quad (3.19)$$

where the second equality expresses the additive decomposition of the deformation gradient tensor into its components due to the rigid-normal motion, \underline{F}_R , and warping deformation, \underline{E}_W . Introducing Eqs.(3.13) and (3.6), the rigid-normal components become

$$\underline{F}_R = \underline{I} + \left[(\underline{\epsilon}_1^* + \eta_3 \underline{Q} \underline{\varkappa}_1^*) / \chi_1 \ (\underline{\epsilon}_2^* + \eta_3 \underline{Q} \underline{\varkappa}_2^*) / \chi_2 \ 0 \right] = \underline{I} + \underline{E}_R^*, \quad (3.20)$$

where $\underline{\underline{I}}$ is the identity matrix, matrix $\underline{\underline{Q}} = [\bar{i}_1 \ \bar{i}_2 \ 0]$, and arrays $\underline{\underline{\epsilon}}_\alpha^*$ and $\underline{\underline{\kappa}}_\alpha^*$ were defined as

$$\underline{\underline{\epsilon}}_\alpha^* = \underline{\underline{T}}_\alpha^* - \bar{i}_\alpha, \quad (3.21a)$$

$$\underline{\underline{\kappa}}_\alpha^* = \underline{\underline{K}}_\alpha^* - \underline{\underline{k}}_\alpha^\dagger. \quad (3.21b)$$

Arrays $\underline{\underline{\epsilon}}_\alpha^*$ and $\underline{\underline{\kappa}}_\alpha^*$ store the shell's in-plane strain and curvature measures, respectively, resulting from the rigid-normal motion.

The deformation gradient tensor now reduces to $\underline{\underline{F}}^{*\dagger} \approx \underline{\underline{I}} + \underline{\underline{E}}_R^* + \underline{\underline{E}}_W^*$ and the corresponding components of Green-Lagrange's strain tensor, denoted $\underline{\underline{\gamma}}^*$, are

$$\underline{\underline{\gamma}}^* = \frac{1}{2}(\underline{\underline{F}}^{*\dagger T} \underline{\underline{F}}^{*\dagger} - \underline{\underline{I}}) = \frac{1}{2}(\underline{\underline{E}}_R^* + \underline{\underline{E}}_R^{*T}) + \frac{1}{2}(\underline{\underline{E}}_W^* + \underline{\underline{E}}_W^{*T}) + \text{h.o.t.} \quad (3.22)$$

For small strains, Green-Lagrange's strain components are the sum of the strains due to the rigid-normal motion, $\underline{\underline{\gamma}}_R^* = (\underline{\underline{E}}_R^* + \underline{\underline{E}}_R^{*T})/2$, and those resulting from warping deformation, $\underline{\underline{\gamma}}_W^* = (\underline{\underline{E}}_W^* + \underline{\underline{E}}_W^{*T})/2$. It is convenient to collect the strain components in an array, $\underline{\underline{\gamma}}^* = \{\gamma_{11}^*, \gamma_{22}^*, \gamma_{33}^*, \gamma_{23}^*, \gamma_{13}^*, \gamma_{12}^*\}^T$ and Eq. (3.22) then becomes $\underline{\underline{\gamma}}^* = \underline{\underline{\gamma}}_R^* + \underline{\underline{\gamma}}_W^*$.

Equation (3.17a) yields the strains due to warping as $\underline{\underline{\gamma}}_W^* = \underline{\underline{v}}^* + \underline{\underline{B}} \underline{\underline{u}}^*$, where the array of warping derivatives, $\underline{\underline{v}}^*$, is defined as

$$\underline{\underline{v}}^* = \underline{\underline{b}}_1^T \frac{\underline{\underline{u}}_{,1}^*}{a_1} + \underline{\underline{b}}_2^T \frac{\underline{\underline{u}}_{,2}^*}{a_2}, \quad (3.23)$$

and matrices $\underline{\underline{b}}_1$ and $\underline{\underline{b}}_2$ are given by Eq. (3.62). Matrix operator $\underline{\underline{B}}$ is defined as

$$\underline{\underline{B}} = \begin{bmatrix} 0 & -k_{13}^\dagger/\chi_1 & k_{11}^\dagger/\chi_1 \\ k_{23}^\dagger/\chi_2 & 0 & k_{22}^\dagger/\chi_2 \\ 0 & 0 & \partial/\partial\eta_3 \\ 0 & -k_{22}^\dagger/\chi_2 + \partial/\partial\eta_3 & 0 \\ -k_{11}^\dagger/\chi_1 + \partial/\partial\eta_3 & 0 & 0 \\ k_{13}^\dagger/\chi_1 & -k_{23}^\dagger/\chi_2 & 0 \end{bmatrix}. \quad (3.24)$$

With the help of Eqs. (3.25), (3.13), and (3.18), the strains due to the rigid-normal motion become $\underline{\underline{\gamma}}_R^* = \underline{\underline{g}} \underline{\underline{\mathcal{E}}}^*$, where matrix $\underline{\underline{g}}$ is given in Eqs. (3.65) and (3.66) and array $\underline{\underline{\mathcal{E}}}^*$ stores the independent strain measures for the shell, defined as

$$\underline{\underline{\mathcal{E}}}^* = \underline{\underline{t}}_1 \underline{\underline{\mathcal{E}}}_1^* + \underline{\underline{t}}_2 \underline{\underline{\mathcal{E}}}_2^*. \quad (3.25)$$

where $a_\alpha \underline{\mathcal{E}}_\alpha^* = \underline{\mathcal{K}}_\alpha^* - \underline{\mathcal{K}}_\alpha^\dagger$ and matrices $\underline{\mathfrak{t}}_\alpha$, are defined in Eq. (3.63) and (3.64). In summary, Green-Lagrange's strain components are

$$\underline{\gamma}^* = \underline{v}^* + \underline{\underline{B}} \underline{u}^* + \underline{\underline{g}} \underline{\mathcal{E}}^*. \quad (3.26)$$

Taking variation of strain measures and introducing the transpositional relationships (3.9), leads to

$$\delta \underline{\mathcal{E}}^* = \underline{\mathfrak{t}}_1 \delta \underline{\mathcal{E}}_1^* + \underline{\mathfrak{t}}_2 \delta \underline{\mathcal{E}}_2^* = \underline{\mathfrak{t}}_1 (\delta \underline{\mathcal{U}}_{,1}^*/a_1 + \tilde{\mathcal{K}}_1^* \delta \underline{\mathcal{U}}^*) + \underline{\mathfrak{t}}_2 (\delta \underline{\mathcal{U}}_{,2}^*/a_2 + \tilde{\mathcal{K}}_2^* \delta \underline{\mathcal{U}}^*). \quad (3.27)$$

3.2.5 Semi-discretization of the Displacement Field

Shell theory is characterized by two-dimensional differential equations governing the displacement field assumed to be a function of the in-plane variables, η_1 and η_2 , only. In the above paragraphs, the displacement field has been treated as a general vector field depending on three independent variables, η_1 , η_2 , and η_3 . To obtain a two-dimensional formulation, the following semi-discretization of the displacement field is performed,

$$\underline{u}^*(\eta_1, \eta_2, \eta_3) = \underline{\underline{N}}(\eta_3) \hat{\underline{u}}(\eta_1, \eta_2), \quad (3.28)$$

where matrix $\underline{\underline{N}}(\eta_3)$ stores the shape functions used in the discretization and array $\hat{\underline{u}}(\eta_1, \eta_2)$ stores the nodal values of the displacement field. Standard, one-dimensional polynomial shape functions [33] are used here. Assuming that the shell's normal material line has been discretized using n nodes, matrix $\underline{\underline{N}}(\eta_3)$, of size $3 \times 3n$, stores the appropriate combination of shape functions.

Equation (3.28) corresponds to a *semi-discretization* of the problem. The finite element mesh extends over the shell's normal material line only and the nodal values of the displacement components remain functions of the in-plane variables, η_1 and η_2 . The shape functions depend on the through-the-thickness variable only, i.e., $\underline{\underline{N}} = \underline{\underline{N}}(\eta_3)$. The dependency of the nodal displacement components on the in-plane variables is stated explicitly: $\hat{\underline{u}} = \hat{\underline{u}}(\eta_1, \eta_2)$. The semi-discretization procedure leads to a numerical treatment of the solution for variable η_3 whereas the dependency of the solution on variables η_1 and η_2 is treated analytically.

Introducing the discretization into Eq. (3.26) yields the components of Green-Lagrange's strain tensor as

$$\underline{\gamma}^* = \underline{\underline{H}} \hat{\underline{v}} + \underline{\underline{B}} \underline{\underline{N}} \hat{\underline{u}} + \underline{\underline{g}} \underline{\mathcal{E}}^*, \quad (3.29)$$

where array $\hat{\underline{v}}$ stores the nodal values of array \underline{v}^* , and matrix $\underline{\underline{H}}$, of size $6 \times 6n$, stores the corresponding shape functions.

3.3 Governing Equations

After deriving expression for the strain energy and work done by the externally applied loads, the governing equations for the shell will be derived from the principle of virtual work.

3.3.1 Strain Energy Expression

The strain energy stored in the shell is found as

$$A = \frac{1}{2} \int_v \underline{\underline{\gamma}}^{*T} \underline{\underline{\mathcal{D}}}^* \underline{\underline{\gamma}}^* dv = \frac{1}{2} \int_{\eta_1, \eta_2} \mathbf{a} a_1 a_2 d\eta_1 d\eta_2, \quad (3.30)$$

where $\underline{\underline{\mathcal{D}}}^*$ is the 6×6 stiffness matrix for the material resolved in the material basis and $dv = a_1 a_2 \chi_1 \chi_2 d\eta_1 d\eta_2 d\eta_3$ is the differential volume element of the shell. The strain energy density per unit area of the shell was defined as $\mathbf{a} = \int_{\eta_3} \chi_1 \chi_2 \underline{\underline{\gamma}}^{*T} \underline{\underline{\mathcal{D}}}^* \underline{\underline{\gamma}}^* d\eta_3$.

Introducing the discretized components of Green-Lagrange's strain tensor given by Eq. (3.29), the strain energy density becomes

$$\mathbf{a} = \frac{1}{2} (\hat{\mathbf{v}} + \underline{\underline{G}} \underline{\underline{\mathcal{E}}}^*)^T \left[\underline{\underline{M}} (\hat{\mathbf{v}} + \underline{\underline{G}} \underline{\underline{\mathcal{E}}}^*) + \underline{\underline{C}}^T \hat{\mathbf{u}} \right] + \frac{1}{2} \hat{\mathbf{u}}^T \left[\underline{\underline{C}} (\hat{\mathbf{v}} + \underline{\underline{G}} \underline{\underline{\mathcal{E}}}^*) + \underline{\underline{E}} \hat{\mathbf{u}} \right], \quad (3.31)$$

where matrix $\underline{\underline{G}}$ stacks the rows of matrix $\underline{\underline{g}}$ for each of the nodes, and matrices $\underline{\underline{M}}$, $\underline{\underline{C}}$, and $\underline{\underline{E}}$ are defined as

$$\underline{\underline{M}} = \int_{\eta_3} \underline{\underline{H}}^T \underline{\underline{\mathcal{D}}}^* \underline{\underline{H}} \chi_1 \chi_2 d\eta_3, \quad (3.32a)$$

$$\underline{\underline{C}} = \int_{\eta_3} (\underline{\underline{B}} \underline{\underline{N}})^T \underline{\underline{\mathcal{D}}}^* \underline{\underline{H}} \chi_1 \chi_2 d\eta_3, \quad (3.32b)$$

$$\underline{\underline{E}} = \int_{\eta_3} (\underline{\underline{B}} \underline{\underline{N}})^T \underline{\underline{\mathcal{D}}}^* (\underline{\underline{B}} \underline{\underline{N}}) \chi_1 \chi_2 d\eta_3. \quad (3.32c)$$

Given the distribution of material stiffness properties these matrices can be evaluated by integration through the thickness of the shell. Matrices $\underline{\underline{M}}$, $\underline{\underline{C}}$, and $\underline{\underline{E}}$ are of size $6n \times 6n$, $3n \times 6n$, and $3n \times 3n$, respectively.

3.3.2 Rigid-Body Motion

Consider a rigid-body motion of the entire shell written as $\underline{u} = \underline{u}_R - \eta_3 \tilde{\underline{B}}_3 \underline{\phi}_R$, where \underline{u}_R are the components of the rigid-body translation and $\underline{\phi}_R = \{-\phi_{R2}, \phi_{R1}, \phi_{R3}\}$ those of an infinitesimal rigid-body rotation. For convenience, the following rigid-body motion array is introduced $\underline{U}_R^T = \{\underline{u}_R^T, \underline{\phi}_R^T\}$ and the nodal displacements become $\underline{u}^* = \underline{z} \underline{U}_R^*$, where matrix $\underline{z} = [\underline{I}, -\eta_3 \tilde{\underline{t}}_3]$ is of size 3×6 and $\underline{U}_R^* = \underline{C}_R^{-1} \underline{U}_R$ stores the components of the corresponding rigid-body motion resolved in frame \mathcal{F}_R . Because array \underline{U}_R represents a rigid-body motion, its spatial derivatives vanish, $\underline{U}_{R,\alpha} = \underline{0}$, and hence, $\underline{U}_{R,\alpha}^* = -a_\alpha \tilde{\underline{K}}_\alpha^* \underline{U}_R^*$, leading to $\hat{\underline{u}}_{,\alpha} = -a_\alpha \underline{z} \tilde{\underline{K}}_\alpha^* \underline{U}_R^*$, where matrix \underline{z} stacks the rows of matrix \underline{z} for each of the nodes.

If the warping field is selected to be a rigid-body motion, the associated strains must vanish and Eq. (3.29) yields

$$\underline{\gamma}_W^* = \left[-\underline{H} \hat{\underline{b}}_1^T \underline{z} \tilde{\underline{K}}_1^* - \underline{H} \hat{\underline{b}}_2^T \underline{z} \tilde{\underline{K}}_2^* + \underline{B} \underline{N} \underline{z} \right] \underline{U}_R^* = \left[-\underline{H} \underline{G} \underline{K}^* + \underline{B} \underline{N} \underline{z} \right] \underline{U}_R^* = \underline{0}, \quad (3.33)$$

where the second equality results from identities (3.67) and matrix $\underline{K}^* = \underline{t}_1 \tilde{\underline{K}}_1^* + \underline{t}_2 \tilde{\underline{K}}_2^*$. Because the strains must vanish for arbitrary rigid-body motions, the bracketed terms must vanish, $\underline{H} \underline{G} \underline{K}^* = \underline{B} \underline{N} \underline{z}$. Pre-multiplying this result by $\underline{H}^T \underline{D}^*$ and $\underline{N}^T \underline{B}^T \underline{D}^*$, then integrating over the shell's thickness yields two important matrix identities

$$\underline{C}^T \underline{z} = \underline{M} \underline{G} \underline{K}^*, \quad (3.34a)$$

$$\underline{E} \underline{z} = \underline{C} \underline{G} \underline{K}^*. \quad (3.34b)$$

3.3.3 Stress Resultants

Two sets of dual variables, denoted $\hat{\underline{P}}_1$ and $\hat{\underline{P}}_2$, are introduced

$$\hat{\underline{P}}_1 = a_1 \frac{\partial \underline{a}}{\partial \hat{\underline{u}}_{,1}} = \hat{\underline{b}}_1 \hat{\underline{P}}, \quad \hat{\underline{P}}_2 = a_2 \frac{\partial \underline{a}}{\partial \hat{\underline{u}}_{,2}} = \hat{\underline{b}}_2 \hat{\underline{P}}, \quad (3.35)$$

where array $\hat{\underline{P}}$, of size $6n \times 1$, is defined as

$$\hat{\underline{P}} = \underline{M}(\hat{\underline{v}} + \underline{G} \underline{\mathcal{E}}^*) + \underline{C}^T \hat{\underline{u}}, \quad (3.36)$$

and stores the nodal forces, as can be verified by introducing Eqs. (3.32a) and (3.32b) into Eq. (3.36) to find $\hat{\underline{P}} = \int_{\eta_3} \underline{H}^T \underline{D}^* \underline{\gamma}^* \chi_1 \chi_2 d\eta_3 = \int_{\eta_3} \underline{H}^T \underline{\mathcal{T}}^* \chi_1 \chi_2 d\eta_3$. Dual

variables \hat{P}_1 and \hat{P}_2 are now interpreted as the nodal forces acting on faces normal to unit vectors \bar{B}_1 and \bar{B}_2 , respectively, and summing up these forces through the shell's thickness yields the corresponding stress resultants

$$\underline{\mathcal{F}}_1^* = \underline{\mathcal{Z}}^T \hat{P}_1, \quad \underline{\mathcal{F}}_2^* = \underline{\mathcal{Z}}^T \hat{P}_2. \quad (3.37)$$

where $\underline{\mathcal{F}}_1^*$ and $\underline{\mathcal{F}}_2^*$ are the stress resultant per length acting of the faces normal to unit vectors \bar{B}_1 and \bar{B}_2 , respectively. Stress resultants $\underline{\mathcal{F}}_1^*$ and $\underline{\mathcal{F}}_2^*$ are not independent but consist of a subset of the independent stress resultants stored in array $\underline{\mathcal{F}}^*$, defined as

$$\underline{\mathcal{F}}^* = \underline{\mathcal{G}}^T \hat{P}. \quad (3.38)$$

The choice of the independent stress resultants is discussed in Appendix B. It then follows that

$$\underline{\mathcal{F}}_1^* = \underline{\mathcal{t}}_1^T \underline{\mathcal{F}}^*, \quad \underline{\mathcal{F}}_2^* = \underline{\mathcal{t}}_2^T \underline{\mathcal{F}}^*. \quad (3.39)$$

Introducing the nodal forces, Eq. (3.36), then yields

$$\underline{\mathcal{F}}^* = (\underline{\mathcal{G}}^T \underline{\mathcal{M}} \underline{\mathcal{G}}) \underline{\mathcal{E}}^* + (\underline{\mathcal{M}} \underline{\mathcal{G}})^T \hat{v} + (\underline{\mathcal{C}} \underline{\mathcal{G}})^T \hat{u}. \quad (3.40)$$

This equation relate the stress resultants to the strain measures and nodal warping. In the absence of warping, i.e., when $\hat{u} = \hat{v} = 0$, this equation reduces to $\underline{\mathcal{F}}^* = \underline{\mathcal{C}}_{RM}^* \underline{\mathcal{E}}^*$, where $\underline{\mathcal{C}}_{RM}^* = \underline{\mathcal{G}}^T \underline{\mathcal{M}} \underline{\mathcal{G}}$ is the Reissner–Mindlin stiffness matrix. Indeed, Reissner–Mindlin's shell theory postulates that normal material lines remain straight, implying that its displacement field is captured adequately by the rigid-normal motion described by matrix $\underline{\mathcal{G}}$. The last two terms of Eq. (3.40) describe the change in stiffness resulting from warping deformation. If the nodal warping, \hat{u} , and their derivatives, \hat{v} , can be expressed in terms of the stress resultants, an equivalent stiffness matrix of the shell can be found. The determination of the warping field is key to the accurate evaluation of the shell's stiffness matrix.

3.3.4 External Virtual Work

The material line is subjected to surface tractions \underline{q}_t^* and \underline{q}_b^* applied at the top and bottom surfaces of the shell, respectively, and body force $\underline{b}^*(\eta_3)$. The virtual work done by these external loads is

$$\begin{aligned} \delta \mathcal{W}_E &= \int_{\eta_1, \eta_2} \left[\int_{\eta_3} \delta \underline{u}^{*T} \underline{b}^* \chi_1 \chi_2 d\eta_3 + (\delta \underline{u}^{*T} \underline{q}^* \chi_1 \chi_2)|_t + (\delta \underline{u}^{*T} \underline{q}^* \chi_1 \chi_2)|_b \right] a_1 a_2 d\eta_1 d\eta_2 \\ &= \int_{\eta_1, \eta_2} \delta \underline{u}^{*T} \underline{\hat{Q}} a_1 a_2 d\eta_1 d\eta_2. \end{aligned} \quad (3.41)$$

The nodal load vector is found by introducing the discretized displacement field, Eq. (3.28), to find $\hat{\underline{Q}} = \int_{\eta_3} \underline{\underline{N}}^T \underline{\underline{b}}^* \chi_1 \chi_2 d\eta_3 + (\underline{\underline{N}}^T \underline{\underline{q}}^* \chi_1 \chi_2)|_t + (\underline{\underline{N}}^T \underline{\underline{q}}^* \chi_1 \chi_2)|_b$. The virtual work per unit area of the shell is $\delta w_E = \delta \hat{\underline{u}}^T \hat{\underline{Q}}$.

3.3.5 Governing Equations

If the drilling rotation constraint, Eq. (3.10), is enforced via Lagrange multiplier λ , the principle of virtual work states that $\int_{\eta_1, \eta_2} \{\delta \alpha - \delta w_E + \delta[\lambda(T_{12}^* - T_{21}^*)]\} a_1 a_2 d\eta_1 d\eta_2 = 0$. Introducing the strain energy density (3.31) and virtual work done by the external loads (3.41) and integrating by parts leads to

$$\int_{\eta_1, \eta_2} \delta \hat{\underline{u}}^T \left[\underline{\underline{C}}(\hat{\underline{v}} + \underline{\underline{G}} \underline{\underline{\mathcal{E}}}^*) - ((a_2 \hat{\underline{b}}_{\underline{1}} \hat{\underline{P}})_{,1} + (a_1 \hat{\underline{b}}_{\underline{2}} \hat{\underline{P}})_{,1}) / (a_1 a_2) + \underline{\underline{E}} \hat{\underline{u}} \right] + \delta \underline{\underline{\mathcal{E}}}_1^{*T} (\underline{\underline{\mathcal{F}}}_1^* + \lambda \underline{\underline{\mathcal{L}}}_1) + \delta \underline{\underline{\mathcal{E}}}_2^{*T} (\underline{\underline{\mathcal{F}}}_2^* - \lambda \underline{\underline{\mathcal{L}}}_2) + \delta \lambda (T_{12}^* - T_{21}^*) a_1 a_2 d\eta_1 d\eta_2 = 0, \quad (3.42)$$

where the boundary terms are omitted for clarity and $\underline{\underline{\mathcal{L}}}_1 = \{0, 1, 0, 0, 0\}^T$ and $\underline{\underline{\mathcal{L}}}_2 = \{1, 0, 0, 0, 0\}^T$.

In the statement of the principle of virtual work, Eq. (3.42), the variation of warping displacements $\delta \hat{\underline{u}}$ and of rigid-normal motion $\delta \underline{\underline{\mathcal{U}}}^*$, see Eq. (3.27), are not independent because the warping field also includes a rigid-body motion, as discussed Sect. 3.2.3. These variations, however, will be treated as independent and will yield two sets of equilibrium equations: the global equations describing geometrically exact shells and the local equations imposing through-the-thickness equilibrium. These two sets are correct but not independent; integration of the local equilibrium equations through the shell's thickness yields the global equilibrium equations.

The local equilibrium equations are obtained by imposing the vanishing of the coefficients of variation $\delta \hat{\underline{u}}$ in the principle of virtual work (3.42), leading to

$$\frac{(a_2 \hat{\underline{b}}_{\underline{1}} \hat{\underline{P}})_{,1}}{a_1 a_2} + \frac{(a_1 \hat{\underline{b}}_{\underline{2}} \hat{\underline{P}})_{,2}}{a_1 a_2} - \underline{\underline{C}} \underline{\underline{G}} \underline{\underline{\mathcal{E}}}^* - \frac{\underline{\underline{C}} \hat{\underline{b}}_{\underline{1}} \hat{\underline{w}}_{,1}}{a_1} - \frac{\underline{\underline{C}} \hat{\underline{b}}_{\underline{2}} \hat{\underline{w}}_{,2}}{a_2} - \underline{\underline{E}} \hat{\underline{w}} = -\hat{\underline{Q}}, \quad (3.43)$$

Introducing the nodal forces from Eq. (3.36) and combing the resulting equations with Eq. (3.40) leads to

$$\underline{\underline{M}}_{11} \frac{\underline{\underline{\mathcal{X}}}_{,11}}{a_1^2} + \underline{\underline{M}}_{12} \frac{\underline{\underline{\mathcal{X}}}_{,12}}{a_1 a_2} + \underline{\underline{M}}_{22} \frac{\underline{\underline{\mathcal{X}}}_{,22}}{a_2^2} + \underline{\underline{H}}_{\underline{1}} \frac{\underline{\underline{\mathcal{X}}}_{,1}}{a_1} + \underline{\underline{H}}_{\underline{2}} \frac{\underline{\underline{\mathcal{X}}}_{,2}}{a_2} - \underline{\underline{E}} \underline{\underline{\mathcal{X}}} = -\underline{\underline{\mathcal{Y}}}, \quad (3.44)$$

where Lamé's coefficients, a_α , and curvatures, k_α^* , are assumed to remain uniform over the shell's midplane surface and arrays $\underline{\underline{\mathcal{X}}}$ and $\underline{\underline{\mathcal{Y}}}$ are defined as

$$\underline{\underline{\mathcal{X}}} = \left\{ \begin{array}{c} \hat{\underline{u}} \\ \underline{\underline{\mathcal{E}}}^* \end{array} \right\}, \quad \underline{\underline{\mathcal{Y}}} = \left\{ \begin{array}{c} \hat{\underline{Q}} \\ \underline{\underline{\mathcal{F}}}^* \end{array} \right\}. \quad (3.45)$$

Linear system (3.44) is a hybrid system that combines the local equilibrium equations expressed in terms the warping field with the global constitutive equation written in terms stress resultants and strain measures. These combined equations plays a central role because they link the local and global problems in a formal manner. These combined equations are solved in both the dimensional reduction and stress recovery processes.

Linear system (3.44) features $N_T = 3n + N_\mathcal{E}$ unknowns, where $N_\mathcal{E}$ is the number of independent strain measures, see Eq. (3.61). The following matrices, each of size $N_T \times N_T$, have been defined

$$\underline{\underline{\check{M}}}_{11} = \begin{bmatrix} \underline{\underline{M}}_{11} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix}, \quad \underline{\underline{\check{M}}}_{12} = \begin{bmatrix} \underline{\underline{M}}_{12} + \underline{\underline{M}}_{12}^T & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix}, \quad \underline{\underline{\check{M}}}_{22} = \begin{bmatrix} \underline{\underline{M}}_{22} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix}, \quad (3.46a)$$

$$\underline{\underline{\check{H}}}_1 = \begin{bmatrix} \underline{\underline{C}}_1^T - \underline{\underline{C}}_1 & \hat{\underline{\underline{b}}}_1 \underline{\underline{M}} \underline{\underline{G}} \\ -(\hat{\underline{\underline{b}}}_1 \underline{\underline{M}} \underline{\underline{G}})^T & \underline{\underline{0}} \end{bmatrix}, \quad \underline{\underline{\check{H}}}_2 = \begin{bmatrix} \underline{\underline{C}}_2^T - \underline{\underline{C}}_2 & \hat{\underline{\underline{b}}}_2 \underline{\underline{M}} \underline{\underline{G}} \\ -(\hat{\underline{\underline{b}}}_2 \underline{\underline{M}} \underline{\underline{G}})^T & \underline{\underline{0}} \end{bmatrix}, \quad (3.46b)$$

$$\underline{\underline{\check{E}}} = \begin{bmatrix} \underline{\underline{E}} & \underline{\underline{C}} \underline{\underline{G}} \\ (\underline{\underline{C}} \underline{\underline{G}})^T & \underline{\underline{G}}^T \underline{\underline{M}} \underline{\underline{G}} \end{bmatrix}, \quad (3.46c)$$

where $\underline{\underline{M}}_{11} = \hat{\underline{\underline{b}}}_1 \underline{\underline{M}} \hat{\underline{\underline{b}}}_1^T$, $\underline{\underline{M}}_{12} = \hat{\underline{\underline{b}}}_1 \underline{\underline{M}} \hat{\underline{\underline{b}}}_2^T$, $\underline{\underline{M}}_{22} = \hat{\underline{\underline{b}}}_2 \underline{\underline{M}} \hat{\underline{\underline{b}}}_2^T$, $\underline{\underline{C}}_1 = \underline{\underline{C}} \hat{\underline{\underline{b}}}_1^T$, and $\underline{\underline{C}}_2 = \underline{\underline{C}} \hat{\underline{\underline{b}}}_2^T$. Matrices $\underline{\underline{\check{M}}}_{11}$, $\underline{\underline{\check{M}}}_{12}$, $\underline{\underline{\check{M}}}_{22}$, and $\underline{\underline{\check{E}}}$ are symmetric whereas matrices $\underline{\underline{\check{H}}}_1$ and $\underline{\underline{\check{H}}}_2$ are skew-symmetric.

The global equilibrium equations are obtained by imposing the vanishing of the coefficients of variation $\delta \underline{\underline{U}}^*$ in the principle of virtual work (3.42), to find

$$\frac{(\underline{\underline{\mathcal{F}}}_1^* + \lambda \underline{\underline{\mathcal{L}}}_1)_{,1}}{a_1} - \tilde{\underline{\underline{\mathcal{K}}}_1^{*T}} (\underline{\underline{\mathcal{F}}}_1^* + \lambda \underline{\underline{\mathcal{L}}}_1) + \frac{(\underline{\underline{\mathcal{F}}}_2^* + \lambda \underline{\underline{\mathcal{L}}}_2)_{,2}}{a_2} - \tilde{\underline{\underline{\mathcal{K}}}_2^{*T}} (\underline{\underline{\mathcal{F}}}_2^* - \lambda \underline{\underline{\mathcal{L}}}_2) = -\underline{\underline{f}}^*, \quad (3.47)$$

where array $\underline{\underline{f}}^* = \underline{\underline{G}}^T \hat{\underline{\underline{Q}}}$ represents the contributions of external loads. The six global equilibrium equations (3.47) impose force and moment equilibrium along and about unit vectors $\underline{\underline{B}}_i$, respectively. The last equation, corresponding to rotational equilibrium about $\underline{\underline{B}}_3$, reduces to $\lambda = 0$. This should be expected since constraint (3.10) simply defines the drilling rotation.

Finally, drilling rotation constraint (3.10) is recovered from variation $\delta \lambda$ in the principle of virtual work (3.42).

3.4 Dimensional Reduction

The first goal of this paper is to obtain a set of reduced, two-dimensional equations for shells. Equations (3.47) provide the global equilibrium equations for geometrically exact shells, but constitutive laws are necessary to complete the formulation. These

constitutive laws are of the form $\underline{\mathcal{E}}^* = \underline{\underline{S}}^* \underline{\mathcal{F}}^*$, where $\underline{\underline{S}}^*$ is the shell's compliance matrix.

3.4.1 Particular Solution

First, a particular solution of the three-dimensional problem will be derived. For the unloaded case, $\underline{f}^* = \underline{0}$, global equilibrium equations (3.47) admit the following particular solution

$$\underline{\mathcal{F}}^*(\eta_1, \eta_2) = \underline{\underline{G}}(\eta_1, \eta_2) \underline{\mathcal{F}}_0^*, \quad (3.48)$$

where $\underline{\mathcal{F}}_0^* = \underline{\mathcal{F}}^*(0, 0)$ and matrix $\underline{\underline{G}}$ is given in Eqs. (3.68) and (3.69). This solution corresponds to the case where the in-plane shearing and twisting resultants are constant, while extension, transverse shearing, and bending resultants are cosine and sine waves in the shell's midplane surface.

The right-hand side of system (3.44) now becomes $\underline{\mathcal{Y}} = \underline{\underline{I}} \underline{\underline{G}} \underline{\mathcal{F}}_0^*$, where matrix $\underline{\underline{I}} = [\underline{0}_{N_{\mathcal{E}} \times 3n}, \underline{I}_{N_{\mathcal{E}} \times N_{\mathcal{E}}}]^T$. The particular solution has a similar form, $\underline{\mathcal{X}} = \underline{\underline{X}} \underline{\mathcal{F}}_0^* = \underline{\underline{X}} \underline{\underline{G}} \underline{\mathcal{F}}_0^*$, where matrix $\underline{\underline{X}}^T = [\underline{W}^T, \underline{S}^T]$ stores the warping, \underline{W} , and strain measures, \underline{S} , induced by unit stress resultants, respectively.

Introducing these results into system (3.44) and imposing that it be satisfied for any loading $\underline{\mathcal{F}}_0^*$ leads to the following set of algebraic equations

$$\underline{\underline{M}}_{11} \underline{\underline{X}} \underline{\tilde{T}}_1^2 + (\underline{\underline{M}}_{12} + \underline{\underline{M}}_{12}^T) \underline{\underline{X}} \underline{\tilde{T}}_1 \underline{\tilde{T}}_2 + \underline{\underline{M}}_{22} \underline{\underline{X}} \underline{\tilde{T}}_2^2 + \underline{\underline{H}}_1 \underline{\underline{X}} \underline{\tilde{T}}_1 + \underline{\underline{H}}_2 \underline{\underline{X}} \underline{\tilde{T}}_2 - \underline{\underline{E}} \underline{\underline{X}} = -\underline{\underline{I}}, \quad (3.49)$$

where $a_1 \underline{\tilde{T}}_1 = \underline{\underline{G}}^{-1} \underline{\underline{G}}_{,1}$ and $a_2 \underline{\tilde{T}}_2 = \underline{\underline{G}}^{-1} \underline{\underline{G}}_{,2}$. Explicit expressions of matrices $\underline{\tilde{T}}_1$ and $\underline{\tilde{T}}_2$ appear in Eqs. (3.70) and (3.71). Equation (3.49) split into the following three recursive sets of equations,

$$\underline{\underline{E}} [\underline{\mathcal{X}}_3 \ \underline{\mathcal{X}}_6 \ \underline{\mathcal{X}}_7 \ \underline{\mathcal{X}}_8 \ \underline{\mathcal{X}}_9] = [\underline{\mathcal{I}}_3 \ \underline{\mathcal{I}}_6 \ \underline{\mathcal{I}}_7 \ \underline{\mathcal{I}}_8 \ \underline{\mathcal{I}}_9], \quad \text{if } \rho_1 \neq \rho_2, \quad (3.50a)$$

$$\underline{\underline{E}} [\underline{\mathcal{X}}_3 \ \underline{\mathcal{X}}_6 \ \underline{\mathcal{X}}_7 \ \underline{\mathcal{X}}_8] = [\underline{\mathcal{I}}_3 \ \underline{\mathcal{I}}_6 \ \underline{\mathcal{I}}_7 \ \underline{\mathcal{I}}_8], \quad \text{if } \rho_1 = \rho_2, \quad (3.50b)$$

$$\left[\underline{\underline{K}}_1 + j \underline{\underline{H}}_1 / \rho_1 \right] (\underline{\mathcal{X}}_1 + j \underline{\mathcal{X}}_4) = \left(\underline{\underline{M}}_{11} \underline{\mathcal{X}}_6 / \rho_1 + \underline{\mathcal{I}}_1 \right) + j \left(\underline{\underline{H}}_1 \underline{\mathcal{X}}_6 + \underline{\mathcal{I}}_4 \right), \quad (3.50c)$$

$$\left[\underline{\underline{K}}_2 + j \underline{\underline{H}}_2 / \rho_2 \right] (\underline{\mathcal{X}}_2 + j \underline{\mathcal{X}}_5) = \left(\underline{\underline{M}}_{22} \underline{\mathcal{X}}_7 / \rho_2 + \underline{\mathcal{I}}_2 \right) + j \left(\underline{\underline{H}}_2 \underline{\mathcal{X}}_7 + \underline{\mathcal{I}}_5 \right), \quad (3.50d)$$

where arrays $\underline{\mathcal{X}}_i$, $\underline{\mathcal{I}}_i$, $i = 1, 2, \dots, N_{\mathcal{E}}$, are the i th column of matrices $\underline{\underline{X}}$ and $\underline{\underline{I}}$, respectively, j is the imaginary unit, $j^2 = -1$, $\underline{\underline{K}}_1 = \underline{\underline{M}}_{11} / \rho_1^2 + \underline{\underline{E}}$, and $\underline{\underline{K}}_2 = \underline{\underline{M}}_{22} / \rho_2^2 + \underline{\underline{E}}$.

3.4.2 Compliance Matrix

Now that a particular solution of system (3.44) has been found, the associated local strain distribution is obtained from Eq. (3.29),

$$\underline{\gamma}^* = \left[\underline{H} \left(\frac{\hat{\underline{b}}_1^T \underline{W} \tilde{\underline{T}}_1}{a_1} + \frac{\hat{\underline{b}}_2^T \underline{W} \tilde{\underline{T}}_2}{a_2} \right) + \underline{B} \underline{N} \underline{W} + \underline{G} \underline{S} \right] \underline{\mathcal{F}}^* = \underline{\mathcal{L}} \underline{\mathcal{F}}^*. \quad (3.51)$$

The corresponding strain energy density is then obtained by integration through the thickness of the shell

$$\alpha = \frac{1}{2} \underline{\mathcal{F}}^{*T} \underline{\mathcal{L}}^T \left[\begin{array}{c} \underline{M} \ \underline{C}^T \\ \underline{C} \ \underline{E} \end{array} \right] \underline{\mathcal{L}} \underline{\mathcal{F}}^* = \frac{1}{2} \underline{\mathcal{F}}^{*T} \underline{S}^* \underline{\mathcal{F}}^*, \quad (3.52)$$

where matrix \underline{S}^* , of size $N_{\mathcal{E}} \times N_{\mathcal{E}}$, is the shell's compliance matrix. The second equality results from an energy equivalence condition: the strain energy computed from the stress resultants should equal that computed based on a particular three-dimensional solution of the problem. The compliance matrix defined by Eq. (3.52) reflects the effects of warping due to material heterogeneity and shell curvature. The constitutive laws now become

$$\underline{\mathcal{E}}^* = \underline{S}^* \underline{\mathcal{F}}^*. \quad (3.53)$$

3.4.3 Discussion

The formulation of geometrically exact shells is now complete. It consists of equilibrium equations (3.47), constitutive laws (3.53), strain-displacement equations (3.25), and the drilling rotation constraint (3.10). They are called “geometrically exact” because they are valid for shells undergoing large motion but small deformation. These equations were first derived by Fox and Simo [32] and similar developments were proposed by Ibrahimbegović [34]. In these references, the shell's normal material line was assumed to remain straight and hence, the stiffness matrix they derived was Reissner–Mindlin's stiffness matrix, $\underline{C}_{RM}^* = \underline{G}^T \underline{M} \underline{G}$, see Eq. (3.40).

Although the present approach gives formally identical equations, important differences exist. Indeed, the proposed developments eliminate the kinematic assumptions associated with commonly used shell theories: the shell's constitutive relationships were derived from three-dimensional elasticity and take into the effect of warping due to material heterogeneity and shell curvature. These effects are reflected in the compliance matrix given by Eq. (3.52). The key feature of the present approach is linear system (3.44) that combines the local equilibrium equations expressed in terms the warping field of with the global constitutive equation written in terms stress resultants and strain measures.

In contrast with higher-order and layer-wise theories, no additional unknowns were introduced in the proposed approach. The computation cost of the present method is identical to that of conventional shell theories. The evaluation of the shell compliance matrix and the stress recovery are pre- and post-processing steps, respectively.

3.5 Stress Recovery

The attention now turns to the recovery of the local stress and strain fields, which depend on the details of the applied loading. The recovery process is based on the two-dimensional shell solutions. In the reduction process, external loads need to be taken into account. The combined local-global equations (3.44) will be solved again to obtain warping and strain measures due to stress resultants and external loads.

In the neighborhood of a point of the shell's midplane surface, the unknowns and right-hand side of equations (3.44) can be expanded as Taylor's series up to order \mathcal{O} ,

$$\underline{\mathcal{X}} = \sum_{k=0}^{\mathcal{O}} \frac{\alpha_1^m \alpha_2^n}{m! n!} \underline{\mathcal{X}}_{m,n}^{(k)}, \quad \underline{\mathcal{Y}}^* = \sum_{k=0}^{\mathcal{O}} \frac{\eta_1^m \eta_2^n}{m! n!} \underline{\mathcal{Y}}_{m,n}^{(k)}. \quad (3.54)$$

where array $\underline{\mathcal{X}}_{m,n}^{(k)T} = \{\hat{\underline{u}}_{m,n}^{(k)T} \underline{\mathcal{E}}_{m,n}^{*(k)T}\}$ stores the coefficients of the expansion of nodal warping and strain measures and array $\underline{\mathcal{Y}}_{m,n}^{(k)T} = \{\hat{\underline{Q}}_{m,n}^{(k)T} \underline{\mathcal{F}}_{m,n}^{*(k)T}\}$ those of the expansion of nodal external loads and stress resultants.

Introducing these expansions into the combined local-global equations (3.44) and matching the coefficients of the monomials leads to the following equations

$$\check{\underline{\underline{E}}}\underline{\mathcal{X}}_{m,n}^{(\mathcal{O})} = \underline{\mathcal{Y}}_{m,n}^{(\mathcal{O})}, \quad (3.55a)$$

$$\check{\underline{\underline{E}}}\underline{\mathcal{X}}_{m,n}^{(\mathcal{O}-1)} = \underline{\mathcal{Y}}_{m,n}^{(\mathcal{O}-1)} + \check{\underline{\underline{H}}}_1 \frac{\underline{\mathcal{X}}_{m+1,n}^{(\mathcal{O})}}{a_1} + \check{\underline{\underline{H}}}_2 \frac{\underline{\mathcal{X}}_{m,n+1}^{(\mathcal{O})}}{a_2}, \quad (3.55b)$$

$$\begin{aligned} \check{\underline{\underline{E}}}\underline{\mathcal{X}}_{m,n}^{(k)} &= \underline{\mathcal{Y}}_{m,n}^{(k)} + \check{\underline{\underline{H}}}_1 \frac{\underline{\mathcal{X}}_{m+1,n}^{(k+1)}}{a_1} + \check{\underline{\underline{H}}}_2 \frac{\underline{\mathcal{X}}_{m,n+1}^{(k+1)}}{a_2} \\ &+ \check{\underline{\underline{M}}}_{11} \frac{\underline{\mathcal{X}}_{m+2,n}^{(k+2)}}{a_1^2} + \check{\underline{\underline{M}}}_{12} \frac{\underline{\mathcal{X}}_{m+1,n+1}^{(k+2)}}{a_1 a_2} + \check{\underline{\underline{M}}}_{22} \frac{\underline{\mathcal{X}}_{m,n+2}^{(k+2)}}{a_2^2}. \end{aligned} \quad (3.55c)$$

Note the recursive nature of the equations where the solutions of systems (3.55a) appear on the right-hand side of systems (3.55b), etc. Equations (3.55a) and (3.55b) represent $\mathcal{O} + 1$ and \mathcal{O} independent linear systems, respectively, for the values of indices m and n such that $m + n = \mathcal{O}$ and $m + n = \mathcal{O} - 1$, respectively. Equation (3.55c) are valid for $k = 0, 1, \dots, \mathcal{O} - 2$; each equations represents a total of $k + 1$ linear systems for $m + n = k$. Because the stress resultants are in equilibrium,

it is proved easily that the solvability conditions are satisfied for each of the systems (3.55).

Once the recursive solution is complete, the corresponding power series expansion of the three-dimensional strains is found

$$\underline{\underline{\gamma}}^* = \underline{\underline{G}} \underline{\underline{\mathcal{E}}}_{m,n}^{*(k)} + \hat{\underline{\underline{b}}}_1^T \frac{\hat{\underline{\underline{u}}}_{m+1,n}^{(k+1)}}{a_1} + \hat{\underline{\underline{b}}}_2^T \frac{\hat{\underline{\underline{u}}}_{m,n+1}^{(k+1)}}{a_2} + \underline{\underline{B}} \underline{\underline{N}} \hat{\underline{\underline{u}}}_{m,n}^{(k)}. \quad (3.56)$$

3.6 Numerical Results

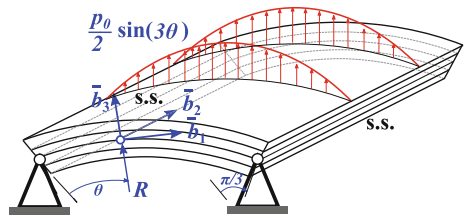
The capabilities of the proposed analysis methodology will be demonstrated by presenting three-dimensional stress and strain fields through the thickness of shells made of anisotropic composite materials and comparing the predictions with exact solutions. The strain energy density computed with the proposed compliance matrix will also be evaluated.

3.6.1 Cylindrical Bending Problem

Figure 3.2 shows a cylindrical composite shell undergoing bending. The midplane of the shell is a cylindrical surface of radius R_m and its width is $L = \pi R_m/3$. The shell is of infinite length along unit vector \bar{b}_2 and is of thickness $h = L/4$. The shell is simply supported along the two edges at $\theta = 0, \pi/3$ and is subjected to distributed transverse pressures $p_t(\theta) = p_b(\theta) = p_0/2 \sin 3\theta$ over the lower and upper surfaces (for clarity, Fig. 3.2 only shows the loading acting over the top surface). The shell consists of 4 plies, each of thickness $t_p = h/4$, all made of the same material with the following stiffness properties: longitudinal, transverse, and shear modulus are $E_L/E_T = 25$, $G_{LT}/E_T = 0.5$, and $G_{TT}/E_T = 0.2$, respectively; Poisson's ratios are $\nu_{LT} = 0.25$ and $\nu_{TN} = 0.25$.

Two lay-up configurations are considered: case (a): $[30^\circ, -30^\circ, -30^\circ, 30^\circ]$ and case (b): $[-60^\circ, -60^\circ, 30^\circ, 30^\circ]$. The lay-ups start with the bottom ply and end with the top ply; 0° fibers are aligned with unit vector \bar{b}_2 and a positive ply angle indicates a right-hand rotation about \bar{b}_3 .

Fig. 3.2 Configuration of the cylindrical bending problem



In the proposed approach, a single four-node one-dimensional element was used to model each ply and the investigation focused on the stress recovery process. Jing and Tzeng [35] obtained an analytical solution of this problem and exact stress resultants and their derivatives were obtained by integrating the exact three-dimensional stress distribution through the shell's thickness.

The distributions of non-dimensional stress components, $\bar{\sigma}_{\alpha\beta} = h^2\sigma_{\alpha\beta}/(p_0L^2)$, $\bar{\sigma}_{\alpha 3} = h\sigma_{\alpha 3}/(p_0L)$, and $\bar{\sigma}_{33} = \sigma_{33}/p_0$, were evaluated through the thickness of the shell at $\eta_1 = \pi R_m/9$. In the stress recovery process, the predictions of four expansion orders, constant, linear, quadratic, and cubic, were compared to assess the convergence of the proposed approach.

Figures 3.3, 3.4, 3.5, 3.6, 3.7 and 3.8 depict the through-the-thickness distribution of non-dimensional stress components. The corresponding results for lay-up (b) are presented in Figs. 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14. In these figures, the predictions for $\mathcal{O} = 0, 1, 2$, and 3 are indicated with symbols \circ, \diamond, \square , and \times , respectively, while the exact solution is indicated by solid lines. For $\mathcal{O} \geq 2$, the predictions of the proposed approach are in close agreement with the exact solutions, demonstrating its good convergence characteristics.

Fig. 3.3 Distribution of stress component $\bar{\sigma}_{11}$, case (a)

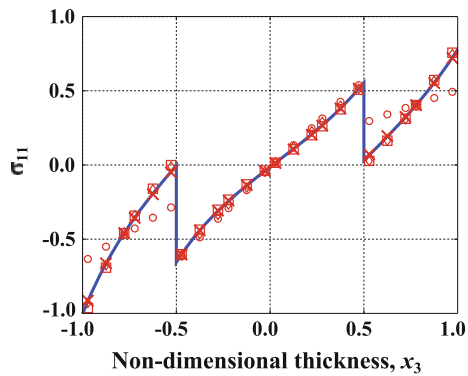


Fig. 3.4 Distribution of stress component $\bar{\sigma}_{22}$, case (a)

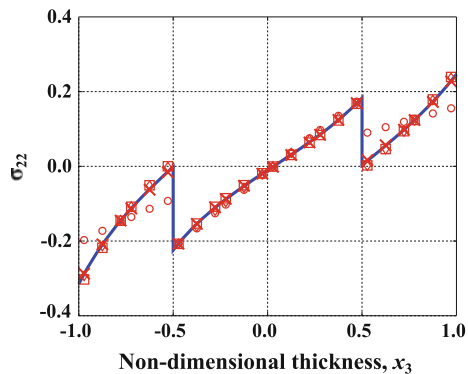


Fig. 3.5 Distribution of stress component $\bar{\sigma}_{12}$, case (a)

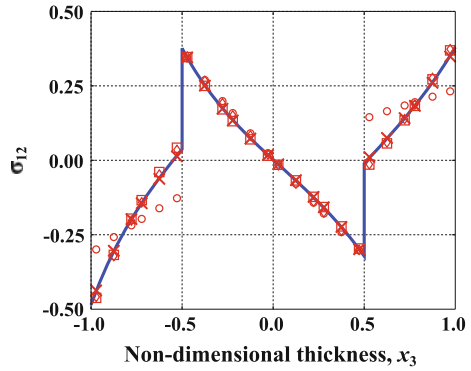


Fig. 3.6 Distribution of stress component $\bar{\sigma}_{33}$, case (a)

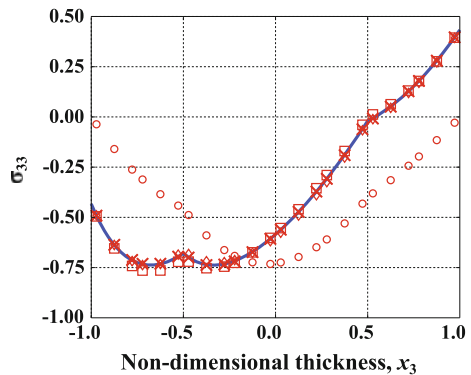
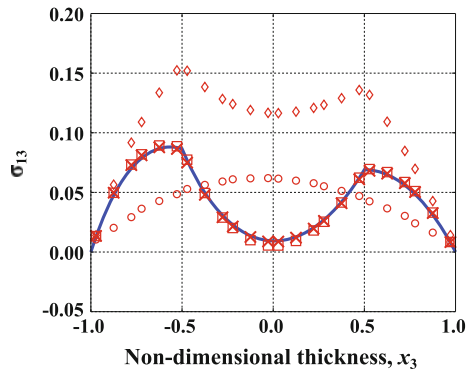


Fig. 3.7 Distribution of stress component, $\bar{\sigma}_{13}$, case (a)



3.6.2 Spatial Bending Problem

The cylindrical composite shell shown in Fig. 3.15 undergoes spatial bending. The midplane of the shell is a cylindrical surface of radius R_m and its length, width, and thickness are $L = R_m$, $b = \pi R_m/3$, and $h = b/10$, respectively. The shell is

Fig. 3.8 Distribution of stress component, $\bar{\sigma}_{23}$, case (a)

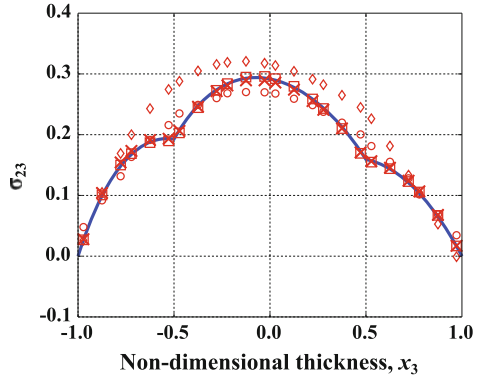


Fig. 3.9 Distribution of stress component $\bar{\sigma}_{11}$, case (b)

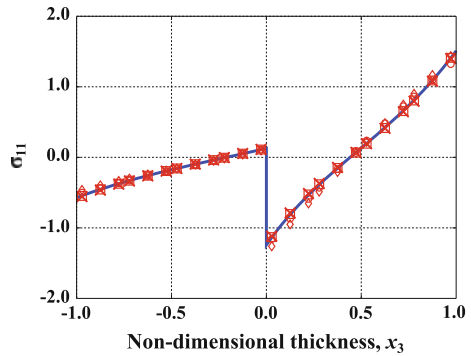
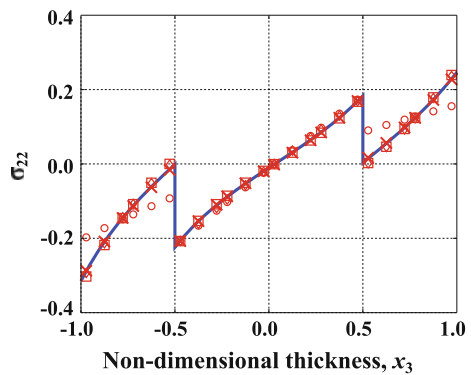


Fig. 3.10 Distribution of stress component $\bar{\sigma}_{22}$, case (b)



simply supported along the four edges at $\theta = 0, \pi/3$ and $\eta_2 = 0, L$. Distributed transverse pressures, $p_t(\eta_1, \eta_2) = p_b(\eta_1, \eta_2) = p_0/2 \sin(\eta_1/b) \sin(\eta_2/L)$ (does not match figure), are applied over both top and bottom surfaces (for clarity, Fig. 3.15 only shows the loading acting over the top surface). The shell consist of 4 plies, each of thickness $t_p = h/4$. All four plies are made of the same material with the following

Fig. 3.11 Distribution of stress component $\bar{\sigma}_{12}$, case (b)

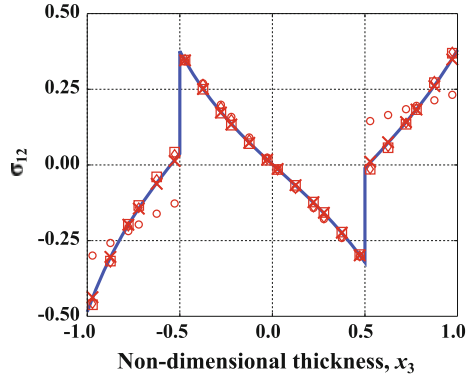


Fig. 3.12 Distribution of stress component $\bar{\sigma}_{33}$, case (b)

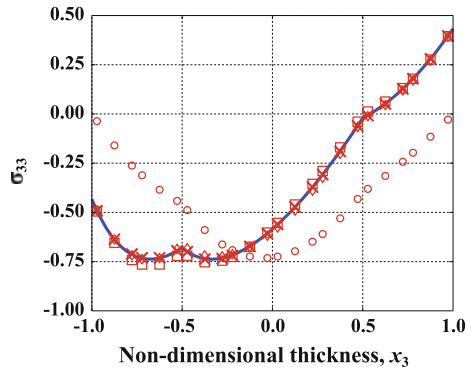
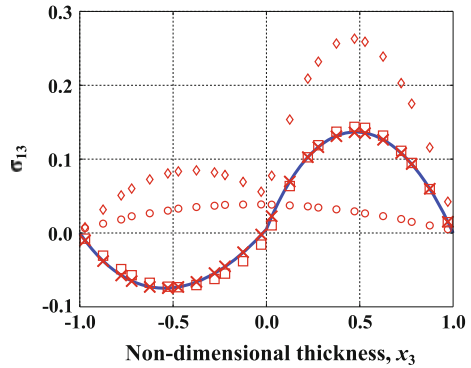


Fig. 3.13 Distribution of stress component, $\bar{\sigma}_{13}$, case (b)



stiffness properties: longitudinal and shear moduli are $E_L/E_T = 25$, $G_{LT}/E_T = 0.5$, and $G_{TT}/E_T = 0.2$, respectively; Poisson’s ratios are $\nu_{LT} = \nu_{TN} = 0.25$.

The lay-up configuration investigated here presents the following stacking sequence, starting from the bottom ply, $[0^\circ, 90_2^\circ, 0^\circ]$; 0° fibers are aligned with unit vector \bar{b}_2 and a positive ply angle indicates a right-hand rotation about unit vector \bar{b}_3 .

Fig. 3.14 Distribution of stress component, $\bar{\sigma}_{23}$, case (b)

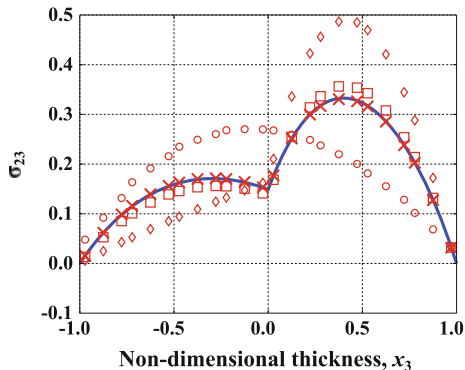
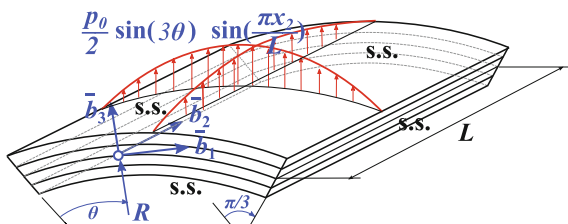


Fig. 3.15 Configuration of the spatial bending problem



In the proposed approach, a single four-node one-dimensional element was used to model each ply. Four sub-matrices of the proposed stiffness matrix, $\underline{C}^* = \underline{S}^{*-1}$, do not vanish: the 3×3 in-plane, 2×2 shearing, 3×4 in-plane/bending coupling, and 4×4 bending stiffness matrices, denoted \underline{A}_{rd} , \underline{S}_{rd} , \underline{C}_{rd} , and \underline{D}_{rd} , respectively. The same holds for the stiffness matrix obtained from the Reissner–Mindlin theory, denoted \underline{C}_{RM}^* . The in-plane stiffness matrices predicted by the two approaches are

$$\frac{\underline{A}_{rd}}{E_T L} = \begin{bmatrix} 1.180 & 0.023 & 0.0 \\ 0.023 & 1.303 & 0.0 \\ 0.0 & 0.0 & 0.050 \end{bmatrix}, \quad \frac{\underline{A}_{RM}}{E_T L} = \begin{bmatrix} 1.314 & 0.0336 & 0.0 \\ 0.0336 & 1.312 & 0.0 \\ 0.0 & 0.0 & 0.050 \end{bmatrix}. \quad (3.57)$$

While the two matrices differ, it is clear that the warping induced deformations taken into account by the present approach have minimal effect on the in-plane stiffness matrix. The same observations apply to the bending stiffness sub-matrix,

$$\frac{\underline{D}_{rd}}{10^{-6} E_T L^3} = \begin{bmatrix} 1806 & -20.75 & 0.0 & 0.0 \\ 20.75 & 334.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 41.73 & 41.67 \\ 0.0 & 0.0 & 41.67 & 41.73 \end{bmatrix}, \quad \frac{\underline{D}_{RM}}{10^{-6} E_T L^3} = \begin{bmatrix} 1849 & 27.96 & 0.0 & 0.0 \\ 27.96 & 340.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 41.73 & 41.67 \\ 0.0 & 0.0 & 41.67 & 41.73 \end{bmatrix}. \quad (3.58)$$

Large discrepancies are observed for the coupling stiffness sub-matrix,

$$\frac{\underline{\underline{C}}_{rd}}{10^{-6}E_T L^3} = \begin{bmatrix} -2243 & -141.5 & 0.0 & 0.0 \\ -523.6 & 327.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & -20.9 & 20.8 \end{bmatrix}, \quad \frac{\underline{\underline{C}}_{RM}}{10^{-6}E_T L^3} = \begin{bmatrix} -1849 & 0.0 & 0.0 & 0.0 \\ 0.0 & 340.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & -20.9 & 20.8 \end{bmatrix}. \tag{3.59}$$

Finally, the same warping deformations affect the shearing stiffness matrix drastically,

$$\frac{\underline{\underline{S}}_{rd}}{10^{-2}E_T L} = \begin{bmatrix} 1.903 & 0.0 \\ 0.0 & 3.179 \end{bmatrix}, \quad \frac{\underline{\underline{S}}_{RM}}{10^{-2}E_T L} = \begin{bmatrix} 3.504 & 0.0 \\ 0.0 & 3.500 \end{bmatrix}. \tag{3.60}$$

As expected, warping induced deformations reduce the shell’s effective stiffness.

Because the configuration of the present problem is simple, Navier type solutions [1] can be found easily using one single sine-wave or cosine-wave term only. Such solutions were obtained based on two stiffness matrices: the stiffness matrix of the proposed approach and that of Reissner–Mindlin’s theory. Table 3.1 lists the magnitudes of the stress resultants and of the mid-point vertical displacement for these two cases. Clearly, the predictions based on the proposed stiffness matrix are in closer agreement with the exact solution of the problem derived by Fan and Zhang [36] than those based on Reissner–Mindlin’s stiffness matrix.

The strain energy density at the point ($\eta_1 = \pi R_m/9, \eta_2 = L/3$) was evaluated based on the exact solution of the problem derived by Fan and Zhang [36] to find $\alpha = 1/2 \int_{\mathcal{L}} \underline{\underline{\gamma}}^{*T} \underline{\underline{D}}^* \underline{\underline{\gamma}}^* d\eta_3 = 0.97969 p_0^2 L/E_T$. Next, the same strain energy was evaluated based on the proposed stiffness matrix and the stress resultants of the

Table 3.1 Comparing the exact and Navier solutions

	Exact	Navier using $\underline{\underline{C}}^*$		Navier using $\underline{\underline{C}}_{RM}^*$	
		Values	Difference (%)	Values	Difference (%)
$N_1^*/(p_0 L)$	0.2727	0.2752	0.94	0.2341	-14.13
$N_2^*/(p_0 L)$	0.3028	0.3051	0.76	0.2727	-9.93
$N_t^*/(p_0 L)$	0.3154	0.3178	0.77	0.2839	-9.97
$Q_1^*/(p_0 L)$	0.1673	0.1672	-0.05	0.1843	10.17
$Q_2^*/(p_0 L)$	0.0718	0.0710	-1.02	0.0678	-5.54
$M_1^*/(p_0 L^2)$	0.0524	0.0524	-0.02	0.0581	10.98
$M_2^*/(p_0 L^2)$	0.0195	0.0193	-1.13	0.0184	-5.97
$M_{12}^*/(p_0 L^2)$	-0.0035	-0.0035	-0.52	-0.0034	-3.04
$M_{21}^*/(p_0 L^2)$	-0.0032	-0.0032	-0.49	-0.0031	-2.47
$E_T u_3^*/(p_0 L^2)$	6.5133	6.5565	0.66	5.8053	-10.87

associated Navier solution, leading to $\alpha = 0.98449 p_0^2 L / E_T$. Finally, when using Reissner–Mindlin’s stiffness matrix and stress resultants of the corresponding Navier solution, the same energy becomes, $\alpha = 0.95962 p_0^2 L / E_T$. The strain energy predicted by the proposed approach is in close agreement with its exact counterpart: the 0.49 % discrepancy is due to discretization and truncation errors. With Reissner–Mindlin’s stiffness matrix, the error becomes 2.0%. Clearly, the stiffness matrix predicted by the proposed approach should be used instead of its Reissner–Mindlin’s counterpart as it captures the strain energy density more accurately.

Next, using the predictions obtained from Navier’s solution with the proposed stiffness matrix, the local stress field at the point $(\eta_1 = \pi R_m / 9, \eta_2 = L / 3)$ was evaluated through the thickness of the shell. In the stress recovery process, the predictions of four expansion orders, constant, linear, quadratic, and cubic, were compared to assess the convergence of the proposed approach.

Figures 3.16, 3.17, 3.18, 3.19, 3.20 and 3.21 depict the through-the-thickness distribution of non-dimensional stress components, $\bar{\sigma}_{ij} = \sigma_{ij} / p_0$. In these figures, the predictions for $\mathcal{O} = 0, 1, 2,$ and 3 are indicated with symbols $\circ, \nabla, \Delta,$ and \times , respectively, while the exact solution by Fan and Zhang [36] is indicated by solid lines. For $\mathcal{O} \geq 2$ the predictions of the proposed approach are in close agreement with

Fig. 3.16 Distribution of stress component $\bar{\sigma}_{11}$

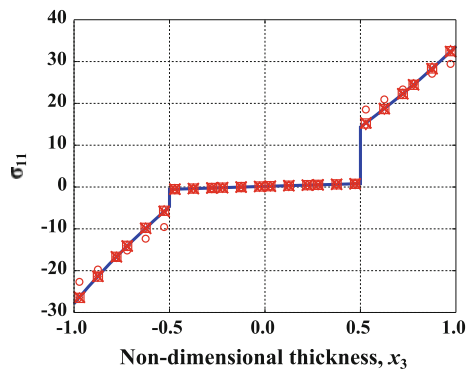


Fig. 3.17 Distribution of stress component $\bar{\sigma}_{22}$

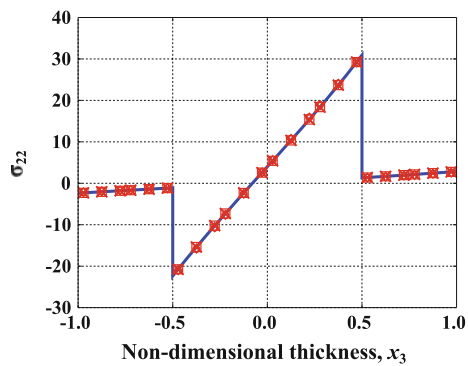


Fig. 3.18 Distribution of stress component $\bar{\sigma}_{33}$

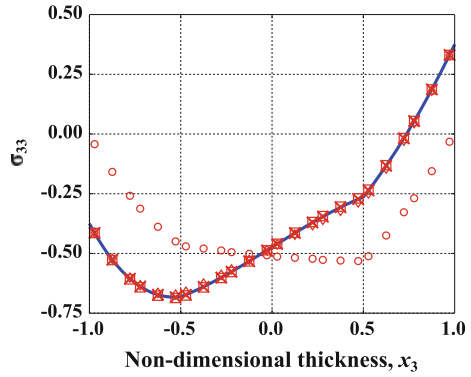


Fig. 3.19 Distribution of stress component $\bar{\sigma}_{12}$

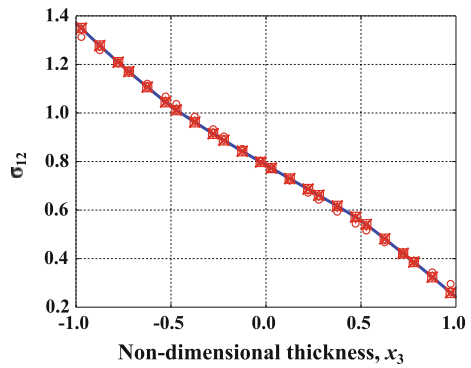
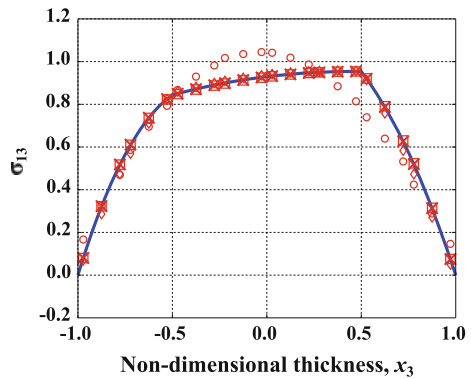
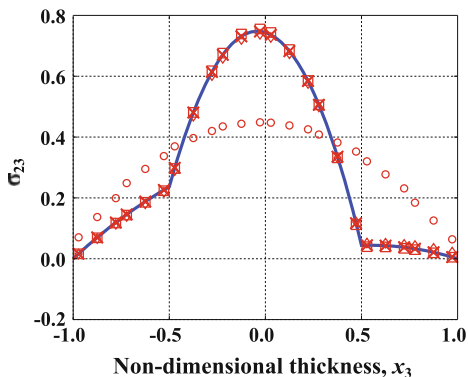


Fig. 3.20 Distribution of stress component $\bar{\sigma}_{13}$



the exact solutions, demonstrating its good convergence characteristics. Although the shell equations were solved using Navier's solution, very accurate stress distributions are recovered through the shell's thickness using $\mathcal{O} \geq 2$.

Fig. 3.21 Distribution of stress component $\bar{\sigma}_{23}$



3.7 Conclusions

This paper has presented a novel approach to the analysis of shells. Kinematically, the problem is decomposed into a rigid-normal motion and a warping field. The strain measures associated with rigid-normal motion and the warping field are assumed to remain small. Consequently, the governing equations of the problem fall into two categories: the equations describing geometrically exact shells and those describing local deformations. The rigid-normal motion of geometrically exact shells is governed by nonlinear, two-dimensional equations, whereas a linear, one-dimensional analysis provides the detailed distribution of three-dimensional stresses and strains.

Numerical examples have been presented to demonstrate the capabilities of the analysis. In the proposed approach, the assumptions associated with commonly used shell theories have been eliminated altogether. Yet, the predicted three-dimensional stress distributions through the shell's thickness compare favorably with exact elasticity solutions. In contrast with higher-order and layer-wise plate theories, the present approach does not increase the number of unknowns used in shell theories. In fact, the proposed approach can be used in conjunction with existing shell models, such as those implemented in multibody analysis or commercial finite element packages. It provides an improved stiffness matrix for the shell and furthermore, accurate, through-the-thickness stress distributions can be obtained through a simple post-processing operation.

Appendix A: Definition of Matrices

Many of the matrices defined in this paper express various quantities in terms of the independent strain measures or stress resultants. The configuration of the shell in its initial configuration affects number of independent parameters. In general, the principal radii of curvature in the initial configuration differ, $\rho_1 \neq \rho_2$, but in case of a spherical shell, $\rho_1 = \rho_2 = \rho$, and hence, $\chi_1 = \chi_2 = \chi$.

For instance, if the radii of curvature differ, the independent strain measures defined by Eq.(3.25) are $\underline{\underline{\mathcal{E}}}^* = \{\epsilon_{11}^*, \epsilon_{22}^*, \epsilon_{12}^* + \epsilon_{21}^*, \epsilon_{13}^*, \epsilon_{23}^*, \varkappa_{11}^*, -\varkappa_{22}^*, -\varkappa_{12}^* + (\epsilon_{21}^* - \epsilon_{12}^*)/(2\rho_1), \varkappa_{21}^* + (\epsilon_{12}^* - \epsilon_{21}^*)/(2\rho_2)\}^T$. On the other hand, if the radii of curvature are equal, $\underline{\underline{\mathcal{E}}}^* = \{\epsilon_{11}^*, \epsilon_{22}^*, \epsilon_{12}^* + \epsilon_{21}^*, \epsilon_{13}^*, \epsilon_{23}^*, \varkappa_{11}^*, -\varkappa_{22}^*, -\varkappa_{12}^* + \varkappa_{21}^*\}^T$. In these expressions, $\epsilon_{\alpha i}^*$ and $\varkappa_{\alpha i}^*$ are the i th components of array $\underline{\underline{\epsilon}}_{\alpha}^*$ and $\underline{\underline{\varkappa}}_{\alpha}^*$, respectively. It is convenient to define

$$N_{\mathcal{E}} = \begin{cases} 9, & \text{if } \rho_1 \neq \rho_2, \\ 8, & \text{if } \rho_1 = \rho_2, \end{cases} \quad (3.61)$$

Array $\underline{\underline{\mathcal{E}}}^*$ is now of size $N_{\mathcal{E}} \times 1$.

Matrices $\underline{\underline{\mathbf{b}}}_1$, and $\underline{\underline{\mathbf{b}}}_2$, of size 3×6 , are defined as

$$\underline{\underline{\mathbf{b}}}_1 = \frac{1}{\chi_1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (3.62a)$$

$$\underline{\underline{\mathbf{b}}}_2 = \frac{1}{\chi_2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (3.62b)$$

Matrices $\hat{\underline{\underline{\mathbf{b}}}}_1$ and $\hat{\underline{\underline{\mathbf{b}}}}_2$ gather the corresponding Boolean matrices at each node.

If the radii of curvature differ, matrices $\underline{\underline{\mathbf{t}}}_1$ and $\underline{\underline{\mathbf{t}}}_2$, of size $N_{\mathcal{E}} \times 6$, are defined as follows

$$\underline{\underline{\mathbf{t}}}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/(2\rho_1) & 0 & -1 & 0 & 0 \\ 0 & 1/(2\rho_2) & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \underline{\underline{\mathbf{t}}}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 1/(2\rho_1) & 0 & 0 & 0 & 0 & 0 \\ -1/(2\rho_2) & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.63)$$

On the other hand, for equal radii of curvature, they become

$$\underline{\underline{\mathbf{t}}}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \quad \underline{\underline{\mathbf{t}}}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (3.64)$$

Next, for differing radii of curvature, matrix $\underline{\underline{g}}$, of size $6 \times N_{\mathcal{E}}$, is defined as

$$\underline{\underline{g}}(\eta_3) = \frac{1}{\chi_1 \chi_2} \begin{bmatrix} \chi_2 & 0 & 0 & 0 & 0 & \chi_2 \eta_3 & 0 & 0 & 0 \\ 0 & \chi_1 & 0 & 0 & 0 & 0 & \chi_1 \eta_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \chi_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \chi_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \chi_m & 0 & 0 & 0 & 0 & \chi_2 \eta_3 & \chi_1 \eta_3 \end{bmatrix}, \quad (3.65)$$

where $\chi_m = (\chi_1 + \chi_2)/2$. For equal radii of curvature, it becomes

$$\underline{\underline{g}} = \frac{1}{\chi} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \eta_3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \eta_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \eta_3 \end{bmatrix}. \quad (3.66)$$

The following identities hold

$$\underline{\underline{g}} \underline{\underline{t}}_{=1} = \underline{\underline{b}}_{=1}^T \underline{\underline{z}}, \quad \underline{\underline{g}} \underline{\underline{t}}_{=2} = \underline{\underline{b}}_{=2}^T \underline{\underline{z}}. \quad (3.67)$$

For differing radii of curvature, matrix $\underline{\underline{\mathcal{G}}}$, of size $N_{\mathcal{E}} \times N_{\mathcal{E}}$, is defined as

$$\underline{\underline{\mathcal{G}}}(\eta_1, \eta_2) = \begin{bmatrix} C_1 & 0 & 0 & -S_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & -S_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_1 & 0 & 0 & C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 & C_2 & 0 & 0 & 0 & 0 \\ (1 - C_1)\rho_1 & 0 & 0 & S_1\rho_1 & 0 & 1 & 0 & 0 & 0 \\ 0 & (1 - C_2)\rho_2 & 0 & 0 & S_2\rho_2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.68)$$

For equal radii of curvature, it becomes

$$\underline{\underline{\mathcal{G}}}(\eta_1, \eta_2) = \begin{bmatrix} C_1 & 0 & 0 & -S_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & -S_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_1 & 0 & 0 & C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 & C_2 & 0 & 0 & 0 & 0 \\ (1 - C_1)\rho & 0 & 0 & S_1\rho & 0 & 1 & 0 & 0 & 0 \\ 0 & (1 - C_2)\rho & 0 & 0 & S_2\rho & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.69)$$

where $C_\alpha = \cos(a_\alpha \eta_\alpha / \rho_\alpha)$ and $S_\alpha = \sin(a_\alpha \eta_\alpha / \rho_\alpha)$.

Finally, for differing radii of curvature, matrices $\tilde{\mathcal{T}}_1$ and $\tilde{\mathcal{T}}_2$, of size $N_{\mathcal{E}} \times N_{\mathcal{E}}$, are defined as

$$\tilde{\mathcal{T}}_1 = \frac{a_1}{\rho_1} \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tilde{\mathcal{T}}_2 = \frac{a_2}{\rho_2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.70)$$

For equal radii of curvature, they become

$$\tilde{\mathcal{T}}_1 = \frac{a_1}{\rho} \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tilde{\mathcal{T}}_2 = \frac{a_2}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.71)$$

Appendix B: Stress Resultants

Stress resultants are integrals of the stress distributions through the thickness of the shell. The force stress resultants are defined as

$$\begin{Bmatrix} N_{11}^* \\ N_{12}^* \\ N_{13}^* \end{Bmatrix} = \int_{\eta_3} \begin{Bmatrix} \sigma_{11}^* \\ \sigma_{12}^* \\ \sigma_{13}^* \end{Bmatrix} \chi_2 d\eta_3, \quad \begin{Bmatrix} N_{21}^* \\ N_{22}^* \\ N_{23}^* \end{Bmatrix} = \int_{\eta_3} \begin{Bmatrix} \sigma_{12}^* \\ \sigma_{22}^* \\ \sigma_{23}^* \end{Bmatrix} \chi_1 d\eta_3, \quad (3.72)$$

where N_{11}^* , N_{12}^* , N_{21}^* , and N_{22}^* are the in-plane forces and N_{13}^* and N_{23}^* the transverse shear forces. The moment stress resultants are defined similarly

$$\begin{Bmatrix} M_{11}^* \\ M_{12}^* \end{Bmatrix} = \int_{\eta_3} \begin{Bmatrix} \sigma_{11}^* \\ \sigma_{12}^* \end{Bmatrix} \eta_3 \chi_1 d\eta_3, \quad \begin{Bmatrix} M_{21}^* \\ M_{22}^* \end{Bmatrix} = \int_{\eta_3} \begin{Bmatrix} \sigma_{12}^* \\ \sigma_{22}^* \end{Bmatrix} \eta_3 \chi_2 d\eta_3, \quad (3.73)$$

where M_{11}^* and M_{22}^* are the bending moments and M_{12}^* and M_{21}^* the twisting moments. The stress resultants acting on faces normal to unit vectors \bar{B}_1 and \bar{B}_2 are then collected into array $\underline{\mathcal{F}}_1^{*T} = \{N_{11}^*, N_{12}^*, N_{13}^*, -M_{12}^*, M_{11}^*, 0\}$ and $\underline{\mathcal{F}}_2^{*T} = \{N_{12}^*, N_{22}^*, N_{23}^*, -M_{22}^*, M_{21}^*, 0\}$, respectively.

If the shell's principal radii of curvature are equal, $\chi_1 = \chi_2$ and hence, $N_{12}^* = N_{21}^*$ and $M_{12}^* = M_{21}^*$. For differing radii of curvature, the nine independent stress resultants are selected as $\underline{\mathcal{F}}^* = \{N_{11}^*, N_{22}^*, (N_{12}^* + N_{21}^*)/2, N_{13}^*, N_{23}^*, M_{11}^*, M_{22}^*, M_{12}^*, M_{21}^*\}$. For equal radii of curvature, the eight independent stress resultants are selected as $\underline{\mathcal{F}}^* = \{N_{11}^*, N_{22}^*, N_{12}^*, N_{13}^*, N_{23}^*, M_{11}^*, M_{22}^*, M_{12}^*\}$. Array $\underline{\mathcal{F}}^*$ is of size $N_{\mathcal{E}} \times 1$. In the selection of the independent stress resultants, the equation of equilibrium about unit vector \bar{B}_3 , $M_{12}^*/\rho_1 - M_{21}^*/\rho_2 + N_{12}^* - N_{21}^* = 0$, is used to eliminate one stress resultant component.

References

1. Timoshenko SP, Woinowsky-Krieger S (1959) Theory of plates and shells. McGraw-Hill Book Company, New York
2. Sanders JL (1959) An improved first-approximation theory for thin shells. Technical report TR R-24, NASA
3. Axelrad EL (1987) Theory of flexible shells. North-Holland, New York
4. Reissner E (1945) The effect of transverse shear deformation on the bending of elastic plates. *Zeitschrift für angewandte Mathematik und Physik* 12:A.69–A.77
5. Reissner E (1947) On bending of elastic plates. *Q Appl Math* 5:55–68
6. Mindlin RD (1951) Influence of rotatory inertia and shear on flexural motions of isotropic elastic plates. *J Appl Mech* 18:31–38
7. Librescu L (1975) Elastostatics and kinetics of anisotropic and heterogeneous shell-type structures. Noordhoff International Publishing, Leyden
8. Whitney JM (1987) Structural analysis of laminated anisotropic plates. Technomic Publishing Company, Lancaster
9. Reddy JN (2004) Mechanics of laminated composite plates and shells: theory and analysis. CRC Press, Boca Raton
10. Christensen RM (1979) Mechanics of composite materials. Wiley, New York
11. Tsai SW, Hahn HT (1980) Introduction to composite materials. Technomic Publishing Co., Inc., Westport
12. Lo KH, Christensen RM, Wu FM (1977) A higher-order theory of plate deformation, II: laminated plates. *J Appl Mech* 44:669–676
13. Reddy JN, Liu CF (1985) A higher-order shear deformation theory of laminated elastic shells. *Int J Eng Sci* 23(3):319–330
14. Pandya BN, Kant T (1988) Higher-order shear deformable theories for flexure of sandwich plates-finite element evaluations. *Int J Solids Struct* 24(12):1267–1286
15. Sciuva MD (1987) An improved shear-deformation theory for moderately thick multilayered anisotropic shells and plates. *J Appl Mech* 54(3):589–596
16. Carrera E (1998) Mixed layer-wise models for multilayered plates analysis. *Compos Struct* 43(1):57–70
17. Noor AK, Burton WS (1989) Assessment of shear deformation theories for multilayered composite plates. *Appl Mech Rev* 42:1–13
18. Friedrichs KO, Dressler RF (1961) A boundary-layer theory for elastic plates. *Commun Pure Appl Math* 14(1):1–33
19. Kalamkarov A (1992) Composite and reinforced elements of construction. Wiley, Chichester
20. Berdichevsky VL (1979) Variational-asymptotic method of shell theory construction. *Prikladnaya Matematika y Mekanika* 43(4):664–687
21. Berdichevsky VL (1982) On the energy of an elastic rod. *Prikladnaya Matematika y Mekanika* 45(4):518–529

22. Sutyryn VG, Hodges DH (1996) On asymptotically correct linear laminated plate theory. *Int J Solids Struct* 33:3649–3671
23. Sutyryn VG (1997) Derivation of plate theory accounting asymptotically correct shear deformation. *J Appl Mech* 64:905–915
24. Yu WB, Hodges DH, Volovoi VV (2002) Asymptotic generalization of Reissner-Mindlin theory: accurate three-dimensional recovery for composite shells. *Comput Methods Appl Mech Eng* 191(44):5087–5109
25. Yu WB, Hodges DH (2004) A geometrically nonlinear shear deformation theory for composite shells. *J Appl Mech* 71(1):1–9
26. Kim JS (2010) An asymptotic analysis of anisotropic heterogeneous plates with consideration of end effects. *J Mech Mater Struct* 4(9):1535–1553
27. Giavotto V, Borri M, Mantegazza P, Ghiringhelli G, Carmaschi V, Maffioli GC, Mussi F (1983) Anisotropic beam theory and applications. *Comput Struct* 16(1–4):403–413
28. Masarati P, Ghiringhelli GL (2005) Characterization of anisotropic, non-homogeneous plates with piezoelectric inclusions. *Comput Struct* 83:1171–1190
29. Gruttmann F, Wagner W (2013) A coupled two-scale shell model with applications to layered structures. *Int J Numer Methods Eng* 94(13):1233–1254
30. Han SL, Bauchau OA (2015) Three-dimensional plate theory. *Comput Struct* (Submitted)
31. Bauchau OA (2011) *Flexible multibody dynamics*. Springer, Dordrecht
32. Fox DD, Simo JC (1992) A drill rotation formulation for geometrically exact shells. *Comput Methods Appl Mech Eng* 98(3):329–343
33. Bathe KJ (1996) *Finite element procedures*. Prentice Hall Inc., Englewood Cliffs
34. Ibrahimbegović A (1994) Stress resultant geometrically nonlinear shell theory with drilling rotations-part I. A consistent formulation. *Comput Methods Appl Mech Eng* 118(3–4):265–284
35. Jing HS, Tzeng KG (1995) Elasticity solution for laminated anisotropic cylindrical panels in cylindrical bending. *Compos Struct* 30(3):307–317
36. Fan JR, Zhang JY (1992) Exact solution for thick laminated shells. *Sci China Ser A* 35(11):632–641

Chapter 4

On the Frictional Contacts in Multibody System Dynamics

Filipe Marques, Paulo Flores and Hamid M. Lankarani

Abstract A comprehensive analysis on the use of different friction force models on the dynamic simulations of multibody mechanical systems is investigated in this work. In this context, some of the most relevant approaches for dealing with friction available in the literature are revisited. In a broad sense, the friction models can be classified into the statics and dynamics models, as they describe the steady-state behavior or utilize extra state variable to capture the dynamic phenomena, respectively. In this process, the main limitations and implications of the friction force models are briefly analyzed. The dynamic responses of a single-mass one degree-of-freedom system with permanent contact, as well as a multibody model of double pendulum impacting the ground at its tip, are examined to analyze and compare the various friction laws. The obtained results suggest that the prediction of the dynamic behavior of multibody systems can strongly depend on the selection of the appropriate friction model as well as frictional parameters.

Keywords Friction · Multibody dynamics · Contact forces · Stick-slip

4.1 Introduction

The dynamic analysis of multibody systems have been increasingly requiring accurate techniques to model the contact interaction between different bodies, in which the evaluation of the generated forces plays a relevant role. This work focuses on different modeling techniques for the friction forces which are generally related

F. Marques (✉) · P. Flores

Department of Mechanical Engineering, School of Engineering,
University of Minho Campus de Azurém, 4804-533 Guimarães, Portugal
e-mail: filipe62055@gmail.com

P. Flores

e-mail: pflores@dem.uminho.pt

H.M. Lankarani

Wichita State University, 1845 N. Fairmount Street, Wichita, KS 67260-0133, USA
e-mail: hamid.lankarani@wichita.edu

to the resistance to the relative motion between contacting surfaces. Friction is a highly complex phenomenon which is often overlooked due to typically lower importance when compared to the normal contact forces. However, friction occurs in all real mechanical systems, even when some techniques are employed to minimize its effects, such as, utilizing a pair of materials with low friction coefficient, improving surface finishing, adding fluid lubricant, or using intermediary elements as bearings. In most of the cases, the presence of friction forces is not desirable due to their dissipative effects and wear production, although sometimes they are preponderant for the desired operation of the system. Thus, the selection of the most appropriate friction force model plays a crucial role in prediction of the dynamic performance of multibody systems.

One of the first studies on friction appeared in the 16th century resulting from Leonardo da Vinci's work, who stated that the friction force is proportional to normal load, opposes to the motion, and is independent of the contact area. This approach was corroborated by Amontons [1]. Coulomb [2] established that friction was independent of velocity magnitude, and developed the first friction model.

Coulomb friction law was the major precursor of the evolution of the friction force models. However, this approach does not allow capturing different friction phenomena which have direct influence on the dynamic response of the system. For instance, this model is not continuous, which can introduce numerical instability during a dynamic simulation. Thus, several friction models have been developed over the years in order to obtain a better agreement with experimental data.

The properties of friction have been intensively studied and several conclusions have been drawn about its main characteristics and dependencies. Some of the issues which must be considered when modelling friction include static friction (stiction), stick-slip, Stribeck effect, viscous friction, frictional lag, pre-sliding displacement and break-away force, etc.

Several experimental works have proved that the friction at rest is higher than the kinetic friction [3–5]. For that reason, the existence of two distinct coefficients of friction is often considered, one for zero velocity and the other for when the relative motion occurs. During the relative movement between different bodies, when the velocity reduces, the friction force increases which may lead to the sticking of the contacting surfaces. This phenomenon is the so-called “stick-slip” [6, 7], which is overcome when the applied force is higher than the static friction force, and consequently, the bodies start sliding. Rabinowicz [4] investigated the essence of the static and kinetic coefficients of friction, and experimentally demonstrated that the coefficient of friction can be described as a function of displacement, with a maximum value for small displacements. This value defines the necessary force to initiate the macroscopic motion and it is often called “break-away” force [4]. The overcoming of this force establishes the boundary between the sticking and sliding regimes. Johannes et al. [8] showed that this force is higher for low force rates. Before reaching the break-away force, the contacting surfaces still have relative motion, which is commonly called “pre-sliding” displacement. This phenomenon consists of a small displacement with an elastic spring behavior which occurs for any external tangential force applied between the contacting bodies [9, 10]. Based on

the adhesion theory, one can state that the asperities of different bodies tend to adhere due to the existence of compressive normal load. Therefore, when a tangential load is applied, the rupture of the bond tends to occur. Before reaching the breaking point, the asperities exhibit elastic, and posteriorly, plastic deformation which causes relative displacement. The sticking regime ends when the external force is high enough to break the junctions of asperities.

The use of two different coefficients of friction (static and kinetic) clearly suggests that friction force is dependent on the relative velocity, for which it can only assume two values. However, Stribeck [11] experimentally demonstrated that the transition between static and kinetic modes is not discrete, but instead a continuous drop of friction force for increasing velocities. This behavior can be noticed mainly for low velocities, and is named “Stribeck effect”.

The friction force presents resistance to the change of friction state, which consists on a delay to the velocity change [12]. This characteristic is frequently called “frictional lag”, and introduces a hysteresis loop between friction force and velocity. Bearing in mind the concept of Stribeck effect, the frictional lag produces lower friction forces when the slip velocity is decreasing and vice-versa. This phenomenon is particularly important for higher accelerations.

Starting from the Coulomb friction law, which is not capable of describing most of the aforementioned effects, a wide range of friction models have been proposed over the last few decades. In general, the models can be divided into three main groups, phenomenological, physics-based and computational. In the former group, the friction models are developed to fit the experimental observations and, therefore, they can capture most of the frictional effects. The physics-based models attempt to describe the interaction between two surfaces through the local physics to derive the frictional behavior which is dependent on the material properties. Finally, several models present discontinuities or singularities which can be a source of numerical instability during the computations. In order to overcome these difficulties, some researchers have been proposing new formulations, which have been shown to be both accurate and efficient from computational point of view.

Another way to classify the friction models consists of dividing them into static and dynamic models [13–16]. The first describes the steady-state behavior of friction, which does not allow for capturing of the friction phenomena entirely. In contrast, the second group uses extra-state variables to increase the complexity and flexibility of the model in order to include the aforementioned phenomena.

Coulomb’s law is widely known and utilized due its simplicity and easiness to implement. Thus, Coulomb’s friction law has been extended and modified by many researchers [13, 17, 18]. In order to include the Stribeck effect, several authors proposed different mathematical expressions to describe that curve [12, 19–22], although the exponential approximation presented by Bo and Pavelescu [23] is the most utilized approach. Karnopp [24] proposed an approach in which the friction force is evaluated as in the sticking phase within a velocity range. This model eliminates the discontinuity at zero velocity, and is capable of capturing the stick-slip motion. In order to overcome some numerical instability of Karnopp model, Leine et al. [25] presented a new model, named switch model. Armstrong-Hélouvy et al. [26] pre-

sented a survey of friction models, and introduced a seven parameter model. More recently, Wojewoda et al. [27] proposed a model that describes the hysteretic behavior of friction, and Awrejcewicz et al. [28] modeled the friction force as a function of the external tangential force for low velocities.

By and large, the dynamic models emerged along with the need of capturing some friction characteristics that the static models were not able to describe. Dahl [29, 30] introduced the first dynamic model based on the stress-strain curve to model the frictional behavior of ball bearings. Comparing with Coulomb friction law, both models do not capture the stick-slip motion. However, the Dahl model presents better response [31, 32], because it is capable of describing the pre-sliding displacement. Most of the dynamic friction models are based on the physical interaction between the surfaces asperities, such as the bristle model [33], the reset integrator [33], the LuGre [34], the Elasto-Plastic [35], among others [36–42]. Generally, these approaches consider an extra state variable related to the bristles deflection.

Thus, the main goal of this work is to present and compare several friction force models that can be utilized in the context of multibody systems formulations. In the sequel of this process, the most relevant static and dynamic friction models are briefly characterized and compared. With the purpose of better identifying the similarities and differences among the several friction models, two examples of application are considered. First, a simple unconstrained one-dimensional mechanical system is considered to illustrate the impact of the using of different friction approaches on the dynamic response. Then, the interaction of a double pendulum with a plane surface is utilized to analyze the behavior of the system with friction models in a non-continuous contact situation.

4.2 Static Friction Models

This section presents several “static” friction force models frequently used in the simulations of multibody mechanical systems. It must be stated that most of the models exhibit a discontinuity of friction force when the relative velocity is zero, which can cause difficulties in describing friction in a realistic manner.

4.2.1 Coulomb Friction

Coulomb presented the first friction model, in which the friction always opposes the relative motion between contacting bodies, and that the magnitude of the friction force is proportional to the normal contact force. This model depends on the relative velocity direction, except for zero velocity where the friction force is a multivalued function of the external tangential force. This model can be described as [2]

$$\mathbf{F} = \begin{cases} F_C \operatorname{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \neq 0 \\ \min(\|\mathbf{F}_e\|, F_C) \operatorname{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| = 0 \end{cases} \quad (4.1)$$

where

$$F_C = \mu_k \|\mathbf{F}_N\| \quad (4.2)$$

in which \mathbf{F}_N is the normal force, F_C is the magnitude of Coulomb friction, μ_k is the kinetic coefficient of friction, \mathbf{F}_e is the external tangential force, and \mathbf{v}_T is the relative velocity of the contacting bodies. This model presents a dependence on the velocity by the *signum* function,

$$\text{sgn}(\mathbf{v}_T) = \begin{cases} \frac{\mathbf{v}_T}{\|\mathbf{v}_T\|} & \text{if } \|\mathbf{v}_T\| \neq 0 \\ \mathbf{0} & \text{if } \|\mathbf{v}_T\| = 0 \end{cases} \quad (4.3)$$

where $\mathbf{0}$ is a null vector with the same dimensions as \mathbf{v}_T . Although this model is quite straightforward to implement, it presents some difficulties since it does not specify a friction force at zero velocity. Thus, this velocity dependence can originate perturbations in the dynamic response of the system. Nevertheless, the Coulomb friction law has been utilized quite often to simulate friction behavior for the sake of simplicity, and since it requires only one input parameter; that is, the coefficient of friction.

4.2.2 Coulomb Model with Stiction

Since the friction force at zero velocity is higher than the kinetic friction, several studies have presented the necessity of introducing a friction model, which includes two different friction coefficients. This modified Coulomb approach has a similar behavior to Coulomb's except in the vicinity of zero velocity. It is also a multivalued function, but can reach a higher friction force, and can be described as follows [14]

$$\mathbf{F} = \begin{cases} F_C \text{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \neq 0 \\ \min(\|\mathbf{F}_e\|, F_S) \text{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| = 0 \end{cases} \quad (4.4)$$

where

$$F_S = \mu_s \|\mathbf{F}_N\| \quad (4.5)$$

in which F_S is the magnitude of static friction, and μ_s is the static coefficient of friction which is higher than the kinetic, μ_k . Although this model considers stiction, in practice, it provides similar behavior when compared to Coulomb's law, with an oscillatory force for low velocities.

4.2.3 Coulomb Model with Viscous Friction

One of most common modification of Coulomb's friction law deals with is the introduction of viscous friction component. This effect can be modeled by using several

approaches, although it is usually considered a linear relation between the relative velocity and the friction force related to the viscosity of surface lubricants. Thus, a friction model with viscous effect can be written as [14]

$$\mathbf{F} = \begin{cases} F_C \operatorname{sgn}(\mathbf{v}_T) + F_v \mathbf{v}_T & \text{if } \|\mathbf{v}_T\| \neq 0 \\ \min(\|\mathbf{F}_e\|, F_C) \operatorname{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| = 0 \end{cases} \quad (4.6)$$

in which F_v is the viscous friction coefficient.

4.2.4 Model with Stribeck Effect

In contrast with the Coulomb model with stiction, the Stribeck effect [11] ensures that the decrease from static to kinetic friction is a continuous process. Thus, the friction force during relative motion is expressed as a continuous function of velocity as

$$\mathbf{F} = \begin{cases} \mathbf{F}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \neq 0 \\ \min(\|\mathbf{F}_e\|, F_S) \operatorname{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| = 0 \end{cases} \quad (4.7)$$

where $\mathbf{F}(\mathbf{v}_T)$ is an arbitrary function that depends on the relative velocity. This model ensures that the friction force diminishes when the relative motion is initiated. Bo and Pavelescu [24] introduced an exponential function, which is widely utilized to describe the Stribeck effect and can be expressed as follows

$$\mathbf{F}(\mathbf{v}_T) = \left(F_C + (F_S - F_C) e^{-\left(\frac{\|\mathbf{v}_T\|}{v_S}\right)^{\delta_\sigma}} \right) \operatorname{sgn}(\mathbf{v}_T) + F_v \mathbf{v}_T \quad (4.8)$$

where v_S is the Stribeck velocity and δ_σ is a factor that relies on the geometry of the contacting surfaces, which is often considered 2 as suggested by Armstrong-Hélouvy [22]. This friction model takes into account the Coulomb, viscous, stiction and Stribeck friction effects. However, it presents the same difficulty as the previous approaches at zero velocity.

4.2.5 Karnopp Model

Since the aforementioned models are multivalued functions for zero velocity, their static behavior cannot be captured during a simulation. To overcome this difficulty, Karnopp [24] proposed a model where the velocity is considered zero, for a specified range. Thus, when the velocities are within this interval, the system's state can change and the model's response will be the same as when the relative velocity is zero. Karnopp model is usually used along with the Coulomb model and can be expressed as

$$\mathbf{F} = \begin{cases} \mathbf{F}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| > D_v \\ \min(\|\mathbf{F}_e\|, F_S) \operatorname{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| \leq D_v \end{cases} \quad (4.9)$$

in which D_v is the tolerance for zero velocity. It is important to select a suitable range of the null velocity. Nevertheless, this zero velocity interval does not comply with the real behavior of the contact.

4.2.6 Threlfall Model

In most of the static models described earlier, the friction force at zero velocity is multivalued, and is evaluated as a function of the external tangential force. In order to simplify and ensure computational efficiency, several authors have proposed alternative methods, which replace the discontinuity at zero velocity by a finite slope model. Thus, Threlfall [17] presented a model that avoids the discontinuity associated with the Coulomb's law, and is written as

$$\mathbf{F} = \begin{cases} F_C \left(1 - e^{-\frac{3\|\mathbf{v}_T\|}{v_0}}\right) \text{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \leq v_0 \\ F_C \text{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| > v_0 \end{cases} \quad (4.10)$$

where v_0 is a specified tolerance velocity. The resemblance with the Coulomb friction law increases with the decreasing of this tolerance velocity.

4.2.7 Bengisu and Akay Model

Bengisu and Akay [43] proposed an approach capable of modeling the Stribeck effect, which can be defined as

$$\mathbf{F} = \begin{cases} \left(-\frac{F_S}{v_0} (\|\mathbf{v}_T\| - v_0)^2 + F_S\right) \text{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| < v_0 \\ (F_C + (F_S - F_C) e^{-\xi(\|\mathbf{v}_T\| - v_0)}) \text{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \geq v_0 \end{cases} \quad (4.11)$$

in which ξ should be a positive parameter representing the negative slope of the sliding state. As with the previous model, when the slope at zero velocity is too large, a small step size is needed to correctly capture the friction for low velocities, which slows down the simulation. In addition, for velocities close to zero, the friction force will always be low, irrelevant of the displacement.

4.2.8 Ambrósio Model

The above mentioned limitations associated with friction force's discontinuity led Ambrósio [18] to propose a modified Coulomb's friction law where the friction force is defined as

$$\mathbf{F} = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{v}_T\| \leq v_0 \\ \frac{\|\mathbf{v}_T\| - v_0}{v_1 - v_0} F_C \operatorname{sgn}(\mathbf{v}_T) & \text{if } v_0 < \|\mathbf{v}_T\| < v_1 \\ F_C \operatorname{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \geq v_1 \end{cases} \quad (4.12)$$

where v_0 and v_1 are the tolerances for the velocity. This approach prevents the friction force from changing direction when the relative velocity is close to zero and, therefore, it eliminates most of the numerical instability. However, it does not describe the stick-slip motion.

4.2.9 Awrejcewicz et al. Model

Awrejcewicz et al. [28] developed a more complete and complex static friction model for dry contact which is dependent of both tangential force and relative velocity. This model is governed by four different equations, one for sliding mode, two for the transition from stick to slip modes, and one for sticking mode, as follows

$$\mathbf{F} = \begin{cases} \mathbf{F}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| > \varepsilon \\ F_S \operatorname{sgn}(\mathbf{F}_e) & \text{if } \|\mathbf{v}_T\| \leq \varepsilon \wedge \|\mathbf{F}_e\| > F_S \wedge \mathbf{F}_e \cdot \mathbf{v}_T \geq 0 \\ (2A - 1) F_S \operatorname{sgn}(\mathbf{v}_T) & \text{if } \|\mathbf{v}_T\| \leq \varepsilon \wedge \|\mathbf{F}_e\| > F_S \wedge \mathbf{F}_e \cdot \mathbf{v}_T < 0 \\ A(-\mathbf{F}_e + F_S \operatorname{sgn}(\mathbf{v}_T)) + \mathbf{F}_e & \text{if } \|\mathbf{v}_T\| \leq \varepsilon \wedge \|\mathbf{F}_e\| \leq F_S \end{cases} \quad (4.13)$$

in which

$$A = \frac{\|\mathbf{v}_T\|^2}{\varepsilon^2} \left(3 - 2 \frac{\|\mathbf{v}_T\|}{\varepsilon} \right)$$

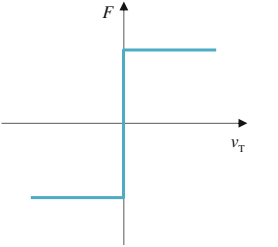
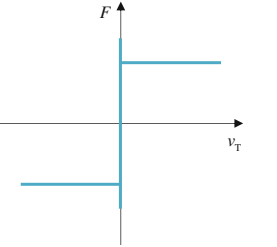
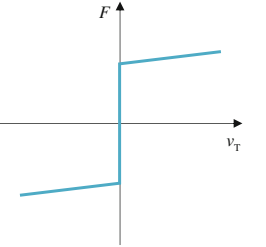
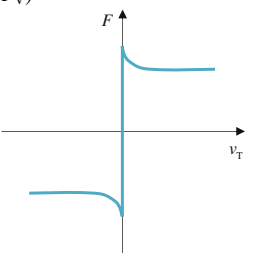
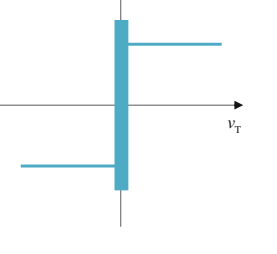
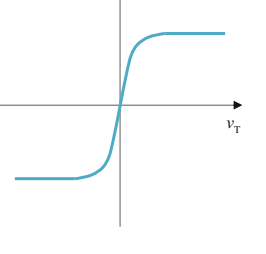
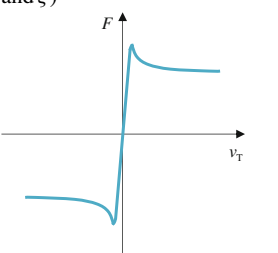
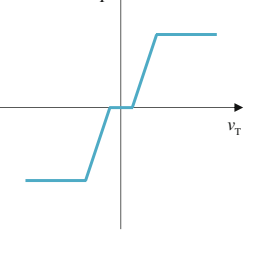
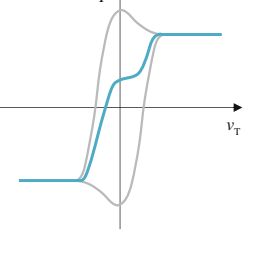
where ε is a velocity tolerance, and $\mathbf{F}(\mathbf{v}_T)$ is an arbitrary friction function for sliding which depends on the velocity. The tolerance velocity defines the limit for sliding state. Below this tolerance, the friction force is also calculated as a function of the external tangential force.

Table 4.1 summarizes the different approaches described earlier for the static friction models. It also includes the model parameters to be used as well as the curves for the friction force versus slip velocity.

4.3 Dynamic Friction Models

This section includes some of the most relevant dynamic friction models. As described earlier, in general, the static friction approaches have limitations in capturing some friction phenomena, such as pre-sliding displacement, or frictional lag. Thus, better alternatives should be examined, namely those available “dynamic friction” models, also named “state variable” models. In a simple manner, the dynamic models use extra state variables together with the velocity to evaluate the friction force.

Table 4.1 Summary of static friction models

Coulomb [2]	Coulomb with stiction [14]	Coulomb w/viscous friction [14]
<p>Parameters: 1 (μ_k)</p> 	<p>Parameters: 2 (μ_k and μ_s)</p> 	<p>Parameters: 2 (μ_k and F_v)</p> 
Model with Stribeck effect [23]	Karnopp [24]	Threlfall [17]
<p>Parameters: 4 (μ_k, μ_s, v_s and F_v)</p> 	<p>Parameters: 3 (μ_k, μ_s and D_v)</p> 	<p>Parameters: 2 (μ_k and v_0)</p> 
Bengisu and Akay [43]	Ambrósio [18]	Awrejcewicz et al. [28]
<p>Parameters: 4 (μ_k, μ_s, v_0 and ξ)</p> 	<p>Parameters: 3 (μ_k, v_0 and v_1)</p> 	<p>Parameters: 3 (μ_k, μ_s and ε)</p> 

4.3.1 Dahl Model

The Dahl friction model [29] was developed with the aim of describing the friction behavior of ball bearings. The basis of this solution is an analogy with the classical stress-strain curve for materials. Dahl observed that in brittle materials, the difference between the stiction and Coulomb friction is difficult to capture. Ductile materials,

however, are more probable of having the stiction behavior, and then exhibiting a decrease in the stress until Coulomb friction is reached. Moreover, the friction force was shown to be dependent on relative velocity and displacement. Dahl model states that when the contacting surfaces are subjected to stress, the friction force increases until rupture occurs. In this context, the stress-strain curve can be described by a differential equation as

$$\frac{dF}{dx} = \sigma \left| 1 - \frac{F}{F_C} \operatorname{sgn}(v_T) \right|^\alpha \operatorname{sgn} \left(1 - \frac{F}{F_C} \operatorname{sgn}(v_T) \right) \quad (4.14)$$

where F denotes the friction force, x is the displacement, F_C is the Coulomb friction, σ represents the stiffness coefficient, and α is a parameter that defines the shape of the material curve. This parameter depends on the material, and usually varies between 0 and 1 for brittle materials, and is higher than 1 for ductile materials. From the analysis of Eq. (4.14), it can be stated that when F tends to F_C , the derivative tends to zero. Thus it can be concluded that the magnitude of the friction force does not exceed F_C .

Equation (4.14) can be transformed into a time derivative, and generalized for the three-dimensional case. Since, typically, α is 1, Eq. (4.14) becomes

$$\frac{d\mathbf{F}}{dt} = \sigma \left(1 - \frac{\mathbf{F} \cdot \operatorname{sgn}(\mathbf{v}_T)}{F_C} \right) \mathbf{v}_T \quad (4.15)$$

Introducing the state variable \mathbf{z} , and assuming that $\mathbf{F} = \sigma \mathbf{z}$, Eq. (4.15) can be written as

$$\frac{d\mathbf{z}}{dt} = \left(1 - \frac{\sigma}{F_C} \mathbf{z} \cdot \operatorname{sgn}(\mathbf{v}_T) \right) \mathbf{v}_T \quad (4.16)$$

It can be observed from Eq. (4.16) that when the system reaches the steady state, the friction force is

$$\mathbf{F} = F_C \operatorname{sgn}(\mathbf{v}_T) \quad (4.17)$$

which is in fact the Coulomb friction model.

It must be highlighted that the Dahl model is not capable of capturing the Stribeck effect and stiction, since it is based on the dry Coulomb friction model with the introduction of pre-sliding displacement through a new state variable, eliminating the discontinuity at zero velocity.

4.3.2 Reset Integrator Model

Haessig and Friedland [33] proposed an evolution of the Dahl model, which considers that the friction force is originated by the elastic and plastic deformations of the surface asperities. Each contact is modeled as a bond between two bristles. The reset

integrator model does not allow for the bond to break, which means that when the strain of a connection increases until reaching the rupture point, the model ensures that it is kept constant. This model uses the average of bristle deflection (\mathbf{z}) to determine the strain in the bond and to account the stiction, as follows

$$\frac{d\mathbf{z}}{dt} = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{z}\| \geq z_0 \wedge \mathbf{z} \cdot \mathbf{v}_T > 0 \\ \mathbf{v}_T & \text{otherwise} \end{cases} \quad (4.18)$$

Similar to other friction models, the reset integrator model is also composed by two state equations, one for sticking mode and another one for sliding mode. The transition between those two phases occurs when the deflection reaches its maximum value z_0 . This friction force can then be defined as follows

$$\mathbf{F} = \begin{cases} \sigma_0(\mathbf{v}_T)(1+a)\mathbf{z} + \sigma_1 \frac{d\mathbf{z}}{dt} & \text{if } \|\mathbf{z}\| < z_0 \\ \sigma_0(\mathbf{v}_T)z_0 \operatorname{sgn}(\mathbf{z}) & \text{if } \|\mathbf{z}\| \geq z_0 \end{cases} \quad (4.19)$$

where $\sigma_1 d\mathbf{z}/dt$ is the damping term that introduces some physical meaning by having damping oscillations and viscous friction effects, a denotes the coefficient pertaining to the stiction, and $\sigma_0(\mathbf{v})$ is the contact stiffness. This friction force model presents a discontinuity when the analysis changes between sticking and sliding situations.

4.3.3 LuGre Model

The LuGre model was originally proposed by Canudas de Wit et al. [34] and can be considered as an extension of the Dahl model [29]. This model is capable of capturing the Stribeck and stiction effects. In a simple way, this model considers friction as the result of the interactions of the surfaces bristles. When a force is applied, the bristles start to deform with spring behavior during the sticking phase. Then if the force is sufficiently large, the bodies start to slip. The model follows as

$$\frac{d\mathbf{z}}{dt} = \left(1 - \frac{\sigma_0}{g(\mathbf{v}_T)} \mathbf{z} \cdot \operatorname{sgn}(\mathbf{v}_T)\right) \mathbf{v}_T \quad (4.20)$$

$$\mathbf{F} = \sigma_0 \mathbf{z} + \sigma_1(\mathbf{v}_T) \frac{d\mathbf{z}}{dt} + f(\mathbf{v}_T) \quad (4.21)$$

where σ_0 is the stiffness of the bristles, $\sigma_1(\mathbf{v})$ is the damping of the bristles which can be set constant, or can be a function of velocity, $f(\mathbf{v})$ is an arbitrary function that describes the viscous effect and $g(\mathbf{v})$ is an arbitrary function that accounts for the Stribeck effect as

$$g(\mathbf{v}_T) = F_C + (F_S - F_C) e^{-\left(\frac{\|\mathbf{v}_T\|}{v_s}\right)^2} \quad (4.22)$$

where F_C is the Coulomb friction, F_S is the static friction and v_S is the characteristic velocity of the Stribeck friction [21]. For $f(\mathbf{v})$, typically a linear viscous friction is considered, that is

$$f(\mathbf{v}_T) = \sigma_2 \mathbf{v}_T \quad (4.23)$$

The effect of this term is quite important when there is a fluid lubricant or when the relative velocities are high.

For a constant velocity, that is, when the system reaches the steady state ($d\mathbf{z}/dt = \mathbf{0}$), the expression to the friction force can be reduced to

$$\mathbf{F} = g(\mathbf{v}_T) \operatorname{sgn}(\mathbf{v}_T) + f(\mathbf{v}_T) \quad (4.24)$$

Thus, considering the Eqs. (4.22) and (4.23), it is possible to conclude that the steady-state values of the LuGre model agrees with the static model defined by Eq. (4.8).

4.3.4 Elasto-Plastic Model

The elasto-plastic model was developed by Dupont et al. [35], and is based on the division of the body displacement into two different components, the elastic and plastic displacement. When the bodies are sticking, the plastic displacement remains constant, while during the sliding phase, the elastic displacement is constant. The friction force can be calculated in a similar way as in the case of LuGre model [34],

$$\mathbf{F} = \sigma_0 \mathbf{z} + \sigma_1 \frac{d\mathbf{z}}{dt} + \sigma_2 \mathbf{v}_T \quad (4.25)$$

where σ_0 is the contact stiffness, σ_1 is the contact damping, and σ_2 is viscosity coefficient. The velocity of bristle deflection is given by

$$\frac{d\mathbf{z}}{dt} = \mathbf{v}_T \left(1 - \alpha(\mathbf{z}, \mathbf{v}_T) \frac{\sigma_0}{g(\mathbf{v}_T)} \mathbf{z} \cdot \operatorname{sgn}(\mathbf{v}_T) \right) \quad (4.26)$$

in which the function $\alpha(\mathbf{z}, \mathbf{v})$ is used to capture stiction, since it just allows elastic displacement until the system reaches the break-away force. This function is expressed as

$$\alpha(\mathbf{z}, \mathbf{v}_T) = \begin{cases} \alpha(\mathbf{z}) & \text{if } \mathbf{v}_T \cdot \mathbf{z} \geq 0 \\ 0 & \text{if } \mathbf{v}_T \cdot \mathbf{z} < 0 \end{cases} \quad (4.27)$$

with

$$\alpha(\mathbf{z}) = \begin{cases} 0 & \text{if } \|\mathbf{z}\| < z_{ba} \\ \frac{1}{2} \left(\sin \left(\pi \frac{\|\mathbf{z}\| - z_{ba}}{z_{max} - z_{ba}} \right) + 1 \right) & \text{if } z_{ba} < \|\mathbf{z}\| < z_{max} \\ 1 & \text{if } z_{max} < \|\mathbf{z}\| \end{cases} \quad (4.28)$$

where z_{\max} is the maximum bristle deflection and z_{ba} is the break-away bristle deflection. The value of z_{\max} can be determined based on the steady state friction, and the relation $z_{ba}/z_{\max} \approx 0.7$ can be used to calculate z_{ba} [35].

4.3.5 Gonthier Model

Gonthier et al. [39] introduced a two-dimensional friction model based on LuGre approach [34]. This approach considers a force from the bending of the bristles, given by

$$\mathbf{F}_{br} = \sigma_0 \mathbf{z} + \sigma_1 \frac{d\mathbf{z}}{dt} \quad (4.29)$$

where σ_0 is the stiffness and σ_1 is the damping coefficient. To ensure a smooth transition between the stick-slip friction regimes, an auxiliary parameter is defined as,

$$s = e^{-\left(\frac{\|\mathbf{v}_T\|}{v_S}\right)^2} \quad (4.30)$$

where v_S is the Stribeck velocity. When the bodies are sticking, the deformation rate will be equal to the relative velocity, while for sliding, the resultant friction force will approach the Coulomb friction force, \mathbf{F}_C .

$$\frac{d\mathbf{z}}{dt} = s \mathbf{v}_T + (1 - s) \left(\frac{1}{\sigma_1} \mathbf{F}_C - \frac{\sigma_0}{\sigma_1} \mathbf{z} \right) \quad (4.31)$$

Coulomb friction has always the velocity direction and can be approximated by

$$\mathbf{F}_C = F_C \text{dir}_\epsilon(\mathbf{v}_T, v_\epsilon) \quad (4.32)$$

where $\text{dir}_\epsilon(\mathbf{v}, v_\epsilon)$ returns the unit vector with velocity direction, and it smooths the vector oscillations for velocities under a certain tolerance, v_ϵ , to diminish the discontinuities in velocity direction. This tolerance velocity is considered $v_\epsilon = 0.01 v_S$.

$$\text{dir}_\epsilon(\mathbf{v}_T, v_\epsilon) = \begin{cases} \frac{\mathbf{v}_T}{\|\mathbf{v}_T\|} & \text{if } \|\mathbf{v}_T\| \geq v_\epsilon \\ \frac{\mathbf{v}_T}{v_\epsilon} \left(\frac{3}{2} \frac{\|\mathbf{v}_T\|}{v_\epsilon} - \frac{1}{2} \left(\frac{\|\mathbf{v}_T\|}{v_\epsilon} \right)^3 \right) & \text{if } \|\mathbf{v}_T\| < v_\epsilon \end{cases} \quad (4.33)$$

This approach includes a temporal lag associated with the dwell-time dependence. To capture that phenomenon, a new state variable is defined as

$$\dot{s}_{dw} = \begin{cases} \frac{1}{\tau_{dw}} (s - s_{dw}) & \text{if } s - s_{dw} \geq 0 \\ \frac{1}{\tau_{dr}} (s - s_{dw}) & \text{if } s - s_{dw} < 0 \end{cases} \quad (4.34)$$

where τ_{dw} is the dwell-time dynamics time constant, and $\tau_{br} = \sigma_1/\sigma_0$ is the bristle dynamics time constant. The time constants should be set according to the desired time delay, a large one for sticking, and a small time delay for sliding. Thus, the maximum friction force can be defined as

$$F_{\max} = F_C + (F_S - F_C) s_{dw} \quad (4.35)$$

where F_C and F_S are the magnitude of Coulomb and static friction, respectively. Thus, the friction force can be expressed as

$$\mathbf{F} = \begin{cases} \mathbf{F}_{br} + \sigma_2 \mathbf{v}_T & \text{if } \|\mathbf{F}_{br}\| \leq F_{\max} \\ F_{\max} \operatorname{sgn}(\mathbf{F}_{br}) + \sigma_2 \mathbf{v}_T & \text{if } \|\mathbf{F}_{br}\| > F_{\max} \end{cases} \quad (4.36)$$

where σ_2 is the viscous damping coefficient. The use of this model results in a set of ordinary differential equations that are quite stiff at low relative velocities and cannot be solved using explicit ODE solvers.

4.3.6 Liang Bristle Model

This bristle friction model [42] is an extension of the model presented in [33] to the three-dimensional space. The average deflection of the bristles is represented by a linear spring, which can stretch and rotate, and it is constrained to the tangential plane of the contact. Thus, for each individual contact, the friction force can be calculated as

$$\mathbf{F} = k_b \mathbf{s} \quad (4.37)$$

where k_b is the bristle stiffness and \mathbf{s} is the average bristle deflection and can be expressed as

$$\mathbf{s}(t) = \begin{cases} \mathbf{s}(t_0) + \int_{t_0}^t \mathbf{v}_T(t) dt & \text{if } \|\mathbf{s}\| < s_{\max} \\ s_{\max} \frac{\mathbf{v}_T}{\|\mathbf{v}_T\|} & \text{if } \|\mathbf{s}\| \geq s_{\max} \end{cases} \quad (4.38)$$

where t_0 is the starting time of the contact, t is the current time and s_{\max} is the maximum bristle deflection that can be defined as

$$s_{\max} = \begin{cases} s_{k \max} = \frac{F_C}{k_b} & \text{if } \|\mathbf{v}_T\| > v_d \\ s_{s \max} = \frac{F_S}{k_b} & \text{if } \|\mathbf{v}_T\| \leq v_d \end{cases} \quad (4.39)$$

where v_d is a threshold velocity, which represents the numerical boundary between the sticking and sliding regimes. Thus, there is a maximum value for bristle deflection for sticking ($s_{s \max}$) and another for sliding ($s_{k \max}$).

The major drawback of this model is related to the transition of the sticking regime to the sliding regime. This is because it corresponds a sudden decrease of the maximum value of the average bristle deflection, which can result in an abrupt change of the friction force. To stabilize the friction force, the Eq.(4.37) can be modified through the introduction of a damping term; i.e.,

$$\mathbf{F} = k_b \mathbf{s} + c_b \dot{\mathbf{s}} \quad (4.40)$$

where c_b is the bristle damping coefficient. When the model reaches the steady-state in the sliding mode, the friction force will be equal to Coulomb friction force. In the sticking mode, the friction force will be higher, and equal to the static friction.

Table 4.2 summarizes a brief comparison among the six different dynamic models, and the friction phenomena that each model captures are listed individually.

Table 4.2 Comparison of dynamic friction models

Model	Features taken into account
Dahl [29]	<ul style="list-style-type: none"> • 3 Parameters (μ_k, σ and α) • Pre-sliding displacement • Coulomb friction
Reset integrator [33]	<ul style="list-style-type: none"> • 4 Parameters (μ_k, μ_s, σ_0 and σ_1) • Pre-sliding displacement • Stiction
LuGre [34]	<ul style="list-style-type: none"> • 6 Parameters (μ_k, μ_s, v_s, σ_0, σ_1 and σ_2) • Pre-sliding displacement • Stiction • Stribeck effect • Viscous friction • Frictional lag
Elasto-plastic [35]	<ul style="list-style-type: none"> • 6 Parameters (μ_k, μ_s, v_s, σ_0, σ_1 and σ_2) • Pre-sliding displacement • Stiction • Stribeck effect • Viscous friction • Frictional lag
Gonthier et al. [39]	<ul style="list-style-type: none"> • 7 Parameters (μ_k, μ_s, v_s, σ_0, σ_1, σ_2 and τ_{dw}) • Pre-sliding displacement • Stiction • Stribeck effect • Viscous friction • Frictional lag
Liang et al. [42]	<ul style="list-style-type: none"> • 5 Parameters (μ_k, μ_s, v_d, k_b and c_b) • Pre-sliding displacement • Stiction

4.4 Comparison of Results for Illustrative Examples

In this section, two demonstrative application examples are considered, namely, the classical 1-DOF spring-mass system and the double pendulum impacting the ground. The main goal of this section is to analyze and discuss the influence of the use of different friction force models on the dynamics response of multibody systems.

4.4.1 Single DOF Mass-Spring Model with Permanent Contact

The classical 1-DOF spring-mass system is utilized as a numerical example of application, which allows for the analysis and comparison of the different friction models (see Fig. 4.1a). This model is widely used as benchmark for validation of friction models [6, 13, 25, 33, 39, 44], and it consists of a block with mass m , which is positioned on a conveyor belt. The belt is moving with a constant velocity. The block is connected by a spring element with stiffness k_s . The system dynamics is governed by the differential equation

$$F = k_s x + m \ddot{x} \quad (4.41)$$

The simulation parameters for the spring-mass system are listed in Table 4.3. The specific parameters associated with each friction model were extracted from the literature and listed in Table 4.4. Initially, the block is located at the origin of the xy coordinate system, and its velocity is the same as that of the belt.

Figure 4.1b–d shows the global results obtained from the simulations with different friction models. In order to keep the analysis simple, the friction approaches are grouped into three classes, namely “static models without stiction”, “static models with stiction”, and “dynamic models”. The behavior of the system is quantified by the plots of the block position, relative velocity and friction force values. The results are shown for 20 s of simulation.

Regarding the static models without stiction, they present a sticking phase related to the initial conditions, since the spring force magnitude is lower than the Coulomb friction force. The fact of having an exact zero relative velocity at the beginning of the simulation avoids any numerical instability for the models with a discontinuity at null

Table 4.3 Simulation parameters for the spring-mass model

Parameter	Value	Parameter	Value
Mass of the block (m)	1 kg	Time step (Δt)	0.00005 s
Velocity of the belt (v_b)	0.1 m/s	Simulation time	20 s
Spring stiffness (k_s)	2 N/m	Integrator algorithm	Runge–Kutta 4th order

Table 4.4 Parameters considered for the different friction models

Parameter	Symbol	Value	Parameter	Symbol	Value
Static coefficient of friction	μ_s	0.15	Tolerance velocity (Awrejcewicz)	ε	0.001 m/s
Kinetic coefficient of friction	μ_k	0.1	Stiffness coefficient	σ	10^5 N/m
Coefficient of viscosity	F_v	0.1 Ns/m	Stiction coefficient	a	0.5
Stribeck velocity	v_s	0.001 m/s	Stiffness coefficient	σ_0	10^5 N/m
Geometry factor	δ_σ	2	Damping coefficient	σ_1	$\sqrt{10^5}$ Ns/m
Tolerance velocity (Karnopp)	D_v	0.001 m/s	Coefficient of viscosity	σ_2	0.1 Ns/m
Tolerance velocity (Threlfall, Bengisu and Akay)	v_0	0.001 m/s	Dwell-time constant	τ_{dw}	2 s
Factor for curve shape	ξ	50 s/m	Bristle stiffness	k_b	50,000 N/m
Tolerance velocity (Ambrósio)	v_0	0.0001 m/s	Threshold velocity	v_d	0.001 m/s
Tolerance velocity (Ambrósio)	v_1	0.001 m/s	Bristle damping	c_b	80 Ns/m

velocity, as in the case of the Coulomb model and the Coulomb with viscous approach. Since these models have a constant Coulomb friction force, or sometimes lower for velocities close to zero, the block exhibits a spring-like behavior. From the analysis of Fig. 4.1b, it can be observed that the results corresponding to the Coulomb with viscous friction shows a distinct behavior, since the viscosity introduces a damping effect in the block oscillations.

For the static models with stiction, the differences are more evident on the systems' response. Both simulations with Coulomb with stiction and Stribeck friction models present numerical instability, as it can be observed in the friction force plot of Fig. 4.1c. This phenomenon is associated with the changes in the velocity direction. The Coulomb model with stiction is capable of reaching the static friction only at the first peak, and because of the initial velocity, it does not stick again. Karnopp and Awrejcewicz models have similar behavior, and present the well-defined stick and slip phases. As the model with Stribeck friction is the only one with viscous friction component, it has faster cycles comparing with the other approaches.

Analyzing the dynamic models, it is possible to observe that Dahl's model is the only one that does not capture the stick-slip phenomenon. From Fig. 4.1d, the remaining models can be divided into two groups, the first includes the LuGre, Elasto-Plastic and Gonthier, and the second one includes the Reset Integrator and Liang. The difference in the dynamics corresponding to these models behavior is caused by presence of a viscous component in the first set of models. Figure 4.2a, b show the plots of the friction force versus displacement and friction force versus relative velocity for the dynamic friction approaches. Comparing these models with

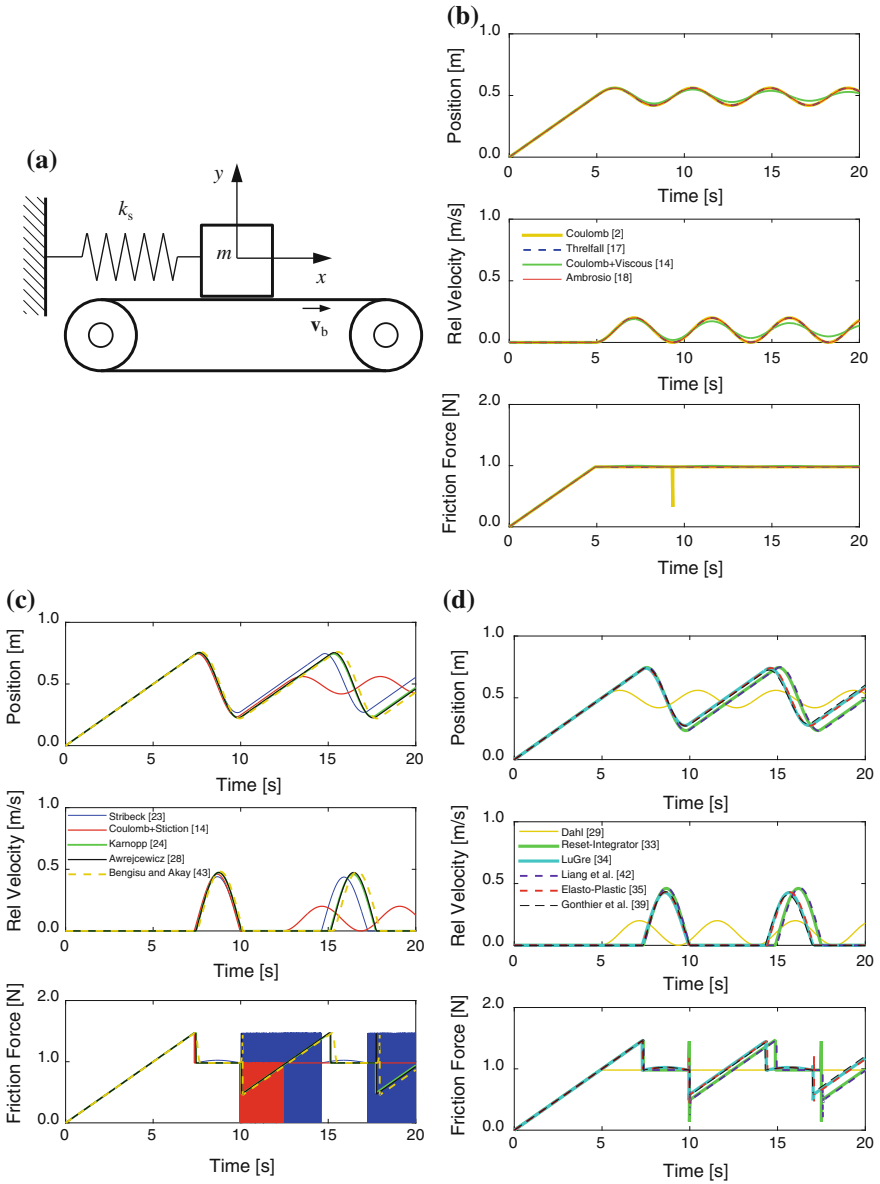


Fig. 4.1 a Representation of the 1-DOF spring-mass model; b static models without stiction; c static models with stiction; d dynamic models

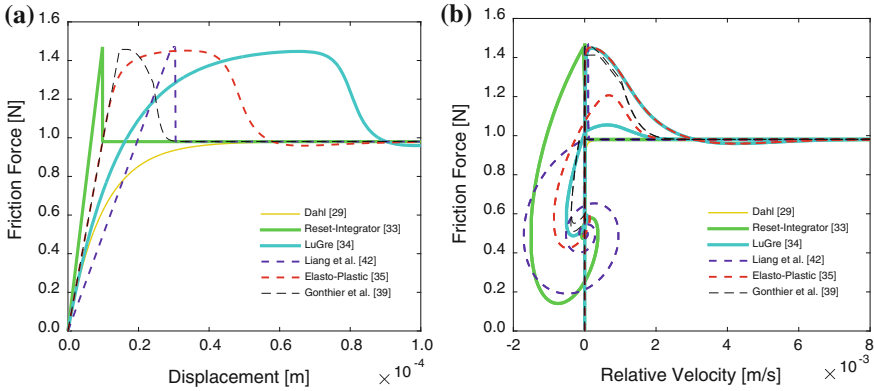


Fig. 4.2 Dynamic models behavior: **a** Friction force versus displacement; **b** Friction force versus relative velocity

Rabinowicz’s work [4], it can be stated that LuGre, Elasto-Plastic and Gonthier’s present the friction force as a function of displacement with a similar shape. During the pre-sliding displacement phase, the slope of the friction force is directly related to the stiffness coefficient of each model (see Fig. 4.2a). In contrast with the static models, the dynamic approaches do not change the force direction with the change of velocity direction, as can be observed in Fig. 4.2b. This behavior is similar to the one obtained in [45]. The Reset Integrator, Elasto-Plastic, and LuGre (to some extent) seem to exhibit an increase of the friction force before sticking. In turn, the Liang’s model shows more instability in velocity, before reaching the sticking phase.

4.4.2 Double Pendulum Impacting Ground

In the previous example, the friction models were analyzed and compared for a system with permanent contact. However, a system with impacts might cause different dynamic responses due to its nonlinear response. In order to study that influence, a double pendulum colliding with the ground is considered here. In Fig. 4.3, the configuration of the double pendulum immediately before the impact is illustrated. The revolute joints were modeled through kinematic constraints [46] and, in the end where the impact occurs, a spherical shape ($r = 0.05$ m) was considered to make the contact detection process easier. Table 4.5 presents the inertia properties and initial conditions of each body used in this example.

The collisions between arm 2 and ground produces high normal force levels. Since the friction force is proportional to the normal load, it is required to have an accurate evaluation of normal forces to perform a correct modelling of friction. For this example, the normal forces are calculated using the compliant force model proposed by Lankarani and Nikravesh [47]. This model is based on Hertzian theory,

Fig. 4.3 Double pendulum configuration

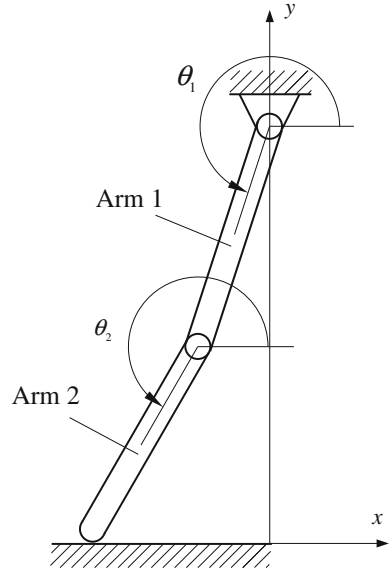


Table 4.5 Properties of the bodies for double pendulum simulation

Parameter	Arm 1	Arm 2
Length (L)	1 m	1 m
Mass (m)	1 kg	1 kg
Moment of inertia (J)	1/12 kg.m ²	1/12 kg.m ²
Angular position (θ)	252°	240°
Angular velocity ($\dot{\theta}$)	1 rad/s	1 rad/s

and takes into account the energy dissipation through the inclusion of a damping term. Thus, the contact force can be written in the following form

$$F_N = K\delta^n \left[1 + \frac{3(1 - c_e^2)}{4} \frac{\dot{\delta}}{\dot{\delta}^{(-)}} \right] \tag{4.42}$$

where K is the contact stiffness, n is an exponent that defines the degree of nonlinearity, c_e is the coefficient of restitution, and $\dot{\delta}^{(-)}$ is the initial impact velocity. The contact stiffness K is evaluated from the material properties of the pendulum and ground to be both steel, and the radius of contact r of arm 2 as

$$K = \frac{2\sqrt{r}}{3 \left(\frac{1-\nu^2}{E} \right)} \tag{4.43}$$

For this model, only four friction force models are considered, namely, the Coulomb law [2], model with Stribeck curve [23], Dahl model [29] and LuGre model [34]. The parameters for friction models are the same used in the Sect. 4.4.1, and are listed in Table 4.4. The properties for the normal contact force model are presented in Table 4.6. The selection of a small time step is required due to the using of dynamic friction models, since they include the resolution of differential equations which need to have small increments.

In this study, the simulations were performed with data provided until the first impact ends. The results are gathered in Table 4.7, which include the impact duration, maximum normal force, maximum friction force, and variation of mechanical energy.

As expected, the results show that the inclusion of friction in the dynamic analysis of the system has a significant effect on its response. The variation of mechanical energy indicates that friction is one of the most important source of impact energy dissipation, since the frictionless impact dissipates almost three times less energy. Another important issue is that including friction has some effect on the normal contact force, while different friction models almost have no influence on the motion in the normal direction. Although the Stribeck model includes static friction which gives a higher maximum friction force compared to Coulomb, it does not affect in the dissipation of energy.

The LuGre model presents more distinct results for all the parameters in Table 4.7 when compared to the remaining models. It is observed that LuGre model is highly affected by the rapid changes of state that happen during impact situations. Thus,

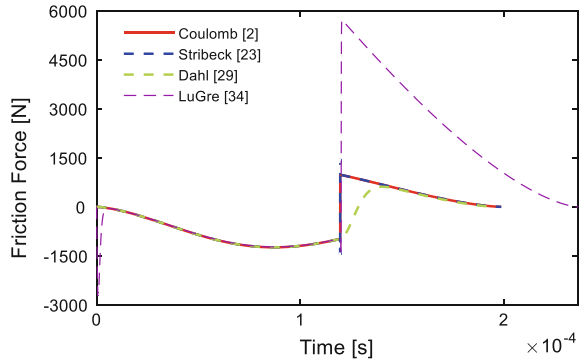
Table 4.6 Parameters used to evaluate normal force model

Parameter	Symbol	Value	Parameter	Symbol	Value
Contact stiffness	K	$3.44 \times 10^{10} \text{ N/m}^{3/2}$	Coefficient of restitution	c_e	0.9
Exponential coefficient	n	1.5	Time step	Δt	$1 \times 10^{-8} \text{ s}$

Table 4.7 Simulation outputs for different friction force models

Model	Impact duration (s)	Maximum normal force (N)	Maximum friction force (N)	Variation of mechanical energy (J)
Frictionless	2.0339×10^{-4}	13,534	0	0.05216
Coulomb	1.9874×10^{-4}	12,395	1239.5	0.14199
Stribeck	1.9874×10^{-4}	12,395	1474.2	0.14199
Dahl	1.9627×10^{-4}	12,395	1239.5	0.14166
LuGre	2.3638×10^{-4}	12,257	5721.9	0.13615

Fig. 4.4 Comparison of friction force for different models in the double pendulum impact



it results in a higher value for the friction force and for the impact duration. The time history of the friction force is represented in Fig. 4.4, and it is again shown that LuGre model has substantial variations. This plot also shows a smooth increase of friction force for the Dahl model.

4.5 Concluding Remarks

A comparative study of some of the most relevant friction force models for multibody systems dynamics has been presented and analyzed in this work. In the sequel of this process, the main characteristics, advantages and limitations of the static and dynamic friction force models were also discussed. To perform a comparative study of the dynamic response of the discussed models, a 1-DOF spring-mass model and a double pendulum with ground impact were utilized as demonstrative examples of application.

The Coulomb friction law was the major precursor of the evolution of the friction force models. This model, however, was shown to have a huge inability in capturing different friction phenomena. These phenomena have a direct influence on the dynamic response of the systems. To mitigate the differences between the reality and friction modeling, robust models must be taken into account to appropriately model general multibody systems.

As it was expected, the friction models present more significant differences at the accelerations level, mainly when there are changes in the velocity direction. Since the main differences between the friction models occurs for small relative velocities, the time integrator and the time step should be carefully chosen in order to be able to correctly capture those differences.

To sum up, the choice of a friction model to implement in a dynamic simulation is not an easy task. Nevertheless, in order to have more complex friction models, it is in general necessary to introduce larger number of parameters to fully define the physics of the friction phenomena. This study illustrates that in order to correctly model, analyze and simulate frictional behavior in multibody systems, an appropriate friction model must be adopted.

References

1. Amontons G (1699) On the resistance originating in machines. *Proc French R Acad Sci* 12:206–222
2. Coulomb CA (1785) *Théorie des machines simples, en ayant égard au frottement de leurs parties, et à la roideur des cordages. Mémoire de Mathématique et de Physique*, Paris, France
3. Morin AJ (1833) New friction experiments carried out at Metz in 1831–1833. *Proc French R Acad Sci* 4:1–128
4. Rabinowicz E (1951) The nature of the static and kinetic coefficients of friction. *J Appl Phys* 22:1373–1379
5. Ścieszka SF, Jankowski A (1996) The importance of static friction characteristics of brake friction couple, and methods of testing. *Tribotest* 3:137–148
6. Rabinowicz E (1956) Stick and slip. *Sci Am* 194:109–118
7. Dieterich J (1978) Time-dependent friction and the mechanics of stick-slip. *Pure Appl Geophys* 116:790–806
8. Johannes VI, Green MA, Brockley CA (1973) The role of the rate of application of the tangential force in determining the static friction coefficient. *Wear* 24:381–385
9. Courtney-Pratt J, Eisner E (1957) The effect of a tangential force on the contact of metallic bodies. *Proc Royal Soc* 238:529–550
10. Hsieh C, Pan Y-C (2000) Dynamic behavior and modelling of the pre-sliding static friction. *Wear* 242:1–17
11. Stribeck R (1902) Die wesentlichen Eigenschaften der Gleitund Rollenlager. *Zeitschrift des Vereines Deutscher Ingenieure* 46:1342–1348, 1432–1438, 1463–1470
12. Hess DP, Soom A (1990) Friction at a lubricated line contact operating at oscillating sliding velocities. *J Tribol* 112:147–152
13. Andersson S, Söderberg A, Björklund S (2007) Friction models for sliding dry, boundary and mixed lubricated contacts. *Tribol Int* 40:580–587
14. Olsson H, Åström KJ, Canudas de Wit C, Gäfvert M, Lischinsky P (1998) Friction models and friction compensation. *Eur J Control* 4:176–195
15. Iurian C, Ikhouane F, Rodellar J, Griñó R (2005) Identification of a system with dry friction. Technical Report, Universitat Politècnica de Catalunya, Spain
16. Marques F, Flores P, Lankarani H (2015) On the frictional contacts in multibody system dynamics. In: *Proceedings of ECCOMAS thematic conference on multibody dynamics*, Barcelona, Spain, 12
17. Threlfall DC (1978) The inclusion of Coulomb friction in mechanisms programs with particular reference to DRAM au programme DRAM. *Mech Mach Theory* 13:475–483
18. Ambrósio JAC (2003) Impact of rigid and flexible multibody systems: deformation description and contact model. *Virtual Nonlinear Multibody Syst* 103:57–81
19. Tustin A (1947) The effects of backlash and of speed-dependent friction on the stability of closed-cycle control systems. *J Inst Electr Eng* 94:143–151

20. Popp K, Stelter P (1990) Nonlinear oscillations of structures induced by dry friction. *Nonlinear Dyn Eng Syst*, pp 233–240
21. Armstrong-Hélouvy B (1991) *Control of machines with friction*. Kluwer Academic Publishers, Norwell
22. Makkar C, Dixon WE, Sawyer WG, Hu G (2005) A new continuously differentiable friction model for control systems design. In: *Proceedings of the 2005 IEEE/ASME, international conference on advanced intelligent mechatronics*, pp 600–605
23. Bo LC, Pavelescu D (1982) The friction-speed relation and its influence on the critical velocity of stick-slip motion. *Wear* 82:277–289
24. Karnopp D (1985) Computer simulation of stick-slip friction in mechanical dynamic systems. *J Dyn Syst Meas Control* 107:100–103
25. Leine RI, van Campen DH, de Kraker A, van den Steen L (1998) Stick-slip vibrations induced by alternate friction models. *Nonlinear Dyn* 16:41–54
26. Armstrong-Hélouvy B, Dupont P, Canudas de Wit C (1994) A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 30:1083–1138
27. Wojewoda J, Stefański A, Wiercigroch M, Kapitaniak T (2008) Hysteretic effects of dry friction: modelling and experimental studies. *Philos Trans Royal Soc A* 366:747–765
28. Awrejcewicz J, Grzelczyk D, Pyryev Y (2008) A novel dry friction modeling and its impact on differential equations computation and Lyapunov exponents estimation. *J Vibroengineering* 10:475–482
29. Dahl PR (1968) A solid friction model, Technical Report, The Aerospace Corporation, El Segundo, California
30. Dahl PR (1976) Solid friction damping in mechanical vibrations. *AIAA J* 14:1675–1682
31. Pennestri E, Valentini PP, Vita L (2007) Multibody dynamics simulation of planar linkages with Dahl friction. *Multibody Syst Dyn* 17:321–347
32. Ksentini O, Abbes MS, Abdessalem J, Chaari F, Haddar M (2012) Study of mass spring system subjected to Dahl friction. *Int J Mech Syst Eng* 2:34–41
33. Haessig DA, Friedland B (1991) On the modeling and simulation of friction. *J Dyn Syst Meas Control* 113:354–362
34. Canudas de Wit C, Olsson H, Åström KJ, Lischinsky P (1995) A new model for control of systems with friction. *IEEE Trans Autom Control* 40:419–425
35. Dupont P, Armstrong B, Hayward V (2000) Elasto-plastic friction model: contact compliance and stiction. *Proc 2000 Am Control Conf* 2:1072–1077
36. Swevers J, Al-Bender F, Ganseman CG, Projogo T (2000) An integrated friction model structure with improved pre-sliding behavior for accurate friction compensation. *IEEE Trans Autom Control* 45:675–686
37. Lampaert V, Al-Bender F, Swevers J (2003) A generalized maxwell-slip friction model appropriate for control purposes. In: *Proceedings of IEEE International conference on physics and control*, St. Petersburg, Russia, pp 1170–1178
38. Al-Bender F, Lampaert V, Swevers J (2004) A novel generic model at asperity level for dry friction force dynamics. *Tribol Lett* 16:81–93
39. Gonthier Y, McPhee J, Lange C, Piedboeuf J-C (2004) A regularized contact model with asymmetric damping and dwell-time dependent friction. *Multibody Syst Dyn* 11:209–233
40. De Moerlooze K, Al-Bender F, Van Brussel H (2010) A generalised asperity-based friction model. *Tribol Lett* 40:113–130
41. Oleksowicz S, Mruk A (2011) A basic theoretical model for friction process at microasperity level. *Tribol Trans* 54:691–700
42. Liang J, Fillmore S, Ma O (2012) An extended bristle friction force model with experimental validation. *Mech Mach Theory* 56:123–137
43. Bengisu MT, Akay A (1994) Stability of friction-induced vibrations in multi-degree-of-freedom systems. *J Sound Vib* 171:557–570
44. Do NB, Ferri AA, Bauchau OA (2007) Efficient simulation of a dynamic system with LuGre friction. *J Comput Nonlinear Dyn* 2:281–289

45. Saha A, Wiercigroch M, Jankowski K, Wahi P, Stefański A (2015) Investigation of two different friction models from the perspective of friction-induced vibrations. *Tribol Int* 90:185–197
46. Nikravesh PE (1988) *Computer-aided analysis of mechanical systems*. Prentice Hall, Englewood Cliffs
47. Lankarani HM, Nikravesh PE (1990) A contact force model with hysteresis damping for impact analysis of multibody systems. *J Mech Des* 112:369–376

Chapter 5

Modeling and Simulation of a 3D Printer Based on a SCARA Mechanism

Eduardo Paiva Okabe and Pierangelo Masarati

Abstract This work presents a dynamic simulation of four arms SCARA (Selective Compliance Articulated Robot for Assembly) mechanism used in 3D printers in an multidisciplinary free software. Different extruder heads, motor supply voltage and microstepping strategies were simulated to show their impact on the construction of the printed part. To do the complete analysis of the printer, it is necessary to simulate the workflow to print a part. The steps of this workflow are part modeling, G-code generation, G-code translation, inverse kinematic analysis, motion translation and dynamic analysis. After accomplishing these steps, the computation of the positioning error completes the analysis. The simulation showed that the microstepping strategy had the greater influence on the construction of the part. The extruder mass became particularly relevant when the voltage was reduced. Simulation of the complete system also showed that electrical and mechanical components can be integrated in one model, although the behavior of components of one domain can restrict the simulation performance of the entire system.

5.1 Introduction

Nowadays companies have been facing shorter and shorter time-to-market cycles. Their success is strictly related to their ability to conceive new products and services. In this environment, the application of new tools to accelerate the innovation process is fundamental to rapidly develop new products. The computational tools for product development linked to the manufacturing design helped to forge the “digital fabrication” concept. The basis of this concept is the application of software for three

E.P. Okabe (✉)

School of Applied Sciences, University of Campinas - UNICAMP,
Rua Pedro Zaccaria, 1300, PO Box 1068, Limeira, Brazil
e-mail: eduardo.okabe@fca.unicamp.br

P. Masarati

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano,
Via La Masa 34, 20156 Milan, Italy
e-mail: pierangelo.masarati@polimi.it

dimensional modeling (CAD), computational simulation (CAE), the code generation (g-code) for CNC machines (CAM—computer aided manufacturing) and rapid prototyping process.

One of the most popular methods of rapid prototyping is the fused deposition method (FDM). It was developed in the end of the 1980s; its basic operation is to add melted plastic directly to the model through an extrusion head. The prototype is built by layers; therefore, the extruder head has to perform a series of repetitive motions until the layer is finished, before moving to the next one. The deposition process has to be relatively fast to assure the adhesion of the new layers on top of the former ones. The control of the motion of the deposition head along with the extrusion speed is fundamental to make the process precise and accurate.

The most common mechanism employed to move the deposition head is the gantry (Cartesian) mechanism. However, other types of mechanism such as Delta (parallel) and SCARA (parallel, redundant, selectively compliant) [6, 20] have been adopted in open source 3D printers because their fabrication and assembly is usually simpler and faster than Cartesian mechanisms. In fact, the latter usually require a more complex support frame. The reduced inertia of the mechanism is another advantage over gantry mechanisms, which makes increased deposition head speed possible and allows the construction of thin wall parts.

The SCARA concept (Selective Compliance Articulated Robot for Assembly) was first introduced in Japan in 1979 [22], and this robot arm was designed to move fast in the horizontal plane with some compliance, essentially lumped in the joints, and with high stiffness regarding vertical motion [21]. It has a small footprint compared to cartesian robots, which renders it very useful to operations in restricted spaces. This kind of mechanism uses vertical axis joints, therefore, the motors do not have to compensate gravity as a Delta geometry robot has to. It can use smaller motors, which minimizes the power needed to hold its position [3].

The multibody simulation of SCARA robots are usually performed as a way to verify their behavior with different setups of actuators and controllers. Padhy [16] developed a dynamic model of a SCARA robot based on Newton–Euler equations. He developed forward recursion equations to compute velocities and accelerations of each link, and backward equations to calculate forces and moments on the end effector. The third link coupled to the end of the second link has a z-translation and a rotation about this axis, therefore, its inertia was added to the second link in order to make the calculations easier.

Fumio [5] developed a simple dynamic model of a SCARA robot with a harmonic drive to estimate the dynamic parameters, and compare the computer model to the experimental setup. The proposed rigid body model estimated the torque within 10% of error; however, this value increased to 30% in lower speed operation, when friction was more significant. Inertia parameters were well estimated, due to the high acceleration which eases the inertia measurement.

Das and Dülger [4] developed a mathematical model of a SCARA robot using a Lagrange formulation of the robot mechanism and a mathematical model of a DC servomotor driven by a PD controller. The simulation was compared to experimental results that demonstrated a close behavior, although the authors did not provide the

error measurement. They applied inverse kinematics to convert the desired trajectory into servomotor rotations.

Pigani and Gallina [17] developed a model of a three link SCARA mechanism actuated by cables. The three link was adopted instead of the two link approach, because the inertia to make small movements is lower, therefore, the tension applied to the cables is also lower. The main problem faced by this kind of configuration is the oscillation, since the mechanism is under-constrained. The mechanism performance improved when damping elements were incorporated in it.

The four arms design (five bar mechanism) as shown in Fig. 5.1 has been chosen to be selectively more rigid and stable compared to the traditional two arms concept, which improves precision on micro movements [24]. Another interesting feature of the four arms design is that both motors can be attached to the support structure, whereas in the traditional two arms configuration one of the motors must be attached to the elbow of the robotic arm.

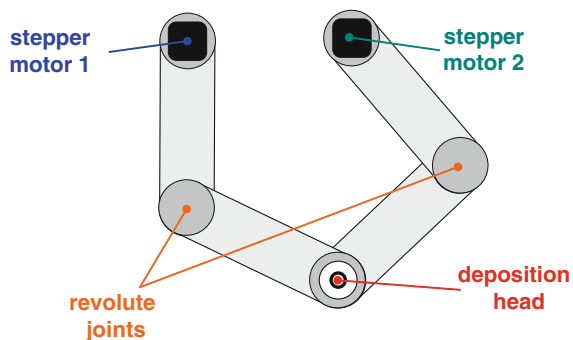
Inverse kinematics plays an important role on the analysis of SCARA mechanism, since the movement of its manipulator has to be translated to motor rotations. It can be applied to model calibration as used by Joubair et al. [8], who developed a kinematic model, where they could identify twelve geometric parameters through the error analysis of five positions. This model was linearized and the Jacobian matrix was employed to calculate the parameters through inverse kinematics.

Another application to the inverse kinematics was presented by Kalra et al. [9], who developed an algorithm to solve the multimodal (multiple configurations) inverse kinematics problem. They applied the simulations to a SCARA robot to identify the optimum strategy, using multiple forward kinematic solutions and a genetic algorithm to optimize the problem. This strategy showed interesting results because it presented approximated solutions, even when the final position was not feasible.

The inverse kinematics of a simple SCARA robot arm was applied in the development of a virtual model by Al Mashhadany [1]. The robot trajectory could be verified through its implementation in a “virtual reality” environment.

Another aspect of SCARA robots that has been investigated in recent years is the motion control. Sprenger et al. [23] developed a SCARA robot to balance an inverted pendulum. The inverse and forward kinematics were formulated in order

Fig. 5.1 Deposition mechanism composed by four arms



to test the control strategy. The control was tested on robot equipped with harmonic drives, and the geometric nonlinearities were calculated by the inverse kinematics, while the other errors were managed by a dynamic compensation.

Hermle and Eberhard [7] proposed a new control strategy and applied to a SCARA robot with one flexible arm. This arm was modeled by finite element and further reduced. The flexible arm was the one of the manipulator, which performs the z-translation. It was incorporated into the multibody model through generalized coordinates. Applying the proposed control strategy with optimized parameters allowed the control of the trajectory and the point-to-point motion.

Visioli and Legnani [26] compared the use of different control strategies to tracking control of the trajectory of a SCARA robot. The single-loop PID controller worked close to the robot resolution in low-speed trajectories. The dynamic model based controller had a better performance in high speed, that can be improved with the addition of a neural network to compensate model errors.

Van Helvoort et al. [25] tested a control of a four arms SCARA robot using a linear fractional transformation applied to an experimental setup. It generated a fourth order model that was validated through a computer simulation and an experiment in time and frequency domains.

Amiri-M et al. [2] proposed a controller of a SCARA robot using a quantitative feedback theory. To test the controller, they developed an inverse kinematics model that was coupled to the dynamic model of the robot. They concluded that one of the problems is associated to the cross coupling effect between joint which certainly makes the movement more nonlinear.

In a 3D printer, the arms are driven by stepper motors, which decreases the need of feedback control, particularly in low speed operation. This represents a cheaper solution compared to servomotor [4] or harmonic drives [5, 23], which are commonly used in industrial robots.

To simulate the behavior of the SCARA mechanism associated to a set of stepper motors, a model was assembled on the free multibody software MBDyn [12]. The motor model proposed by Morar [13] was implemented in a user-defined module and further integrated in the complete mechanism model.

MBDyn features an inverse kinematic and dynamic analysis which is particularly interesting for the study of this kind of mechanism, since it translates the desired trajectory of the deposition head into joint rotations, including the case of redundant mechanisms, providing an estimate of the required torques.

Figure 5.2 shows the workflow of the simulation performed to analyze the mechanism.



Fig. 5.2 Workflow of the dynamic analysis of the mechanism

The first step is to model the part that is going to be fabricated in a solid modeler. Then, a slicing software, such as Slic3r [18], is employed to generate the G-code program. This program tells the printer how it should move to build the part layer by layer. This slicing software takes into consideration some machine limits such as maximum printable dimensions, number of extrusion heads, maximum speed and resolution.

Normally, the firmware of the printer has to translate the desired motions of the deposition head into a sequence of motor steps. In this work, a Python script interprets the G-code generated, and provides the correct displacement, velocity and acceleration of the deposition head to the MBDyn's inverse kinematic analysis.

At the end of the inverse kinematic analysis, a file is generated with the rotation of each arm, which can be directly translated into the motion of the stepper motors. Another script interprets the continuous movement provided by the former analysis into discrete steps that need to be performed by each of the motors. This phase is called motion discretization; it yields a file with time and direction of each step.

The dynamic simulation of the model considers rigid arms, but the stepper motors are modeled to show their influence on the dynamic behavior of the deposition mechanism. Even though the discrete motions are relatively close to the actual trajectory to be followed by the deposition head, the inertia of the components and the stepper motor behavior deviates the motion of the deposition head with respect to the desired path.

5.2 Methodology

The SCARA based printer is a mechanical/electrical system that can be formulated as a system of Differential-Algebraic Equations (DAE) [12]:

$$\begin{aligned} \mathbf{M}\dot{\mathbf{q}} - \boldsymbol{\beta} &= \mathbf{0} \\ \dot{\boldsymbol{\beta}} + \left(\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{q}}\right)^T \boldsymbol{\lambda}_\phi + \left(\frac{\partial\boldsymbol{\psi}}{\partial\dot{\mathbf{q}}}\right)^T \boldsymbol{\lambda}_\psi &= \sum \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \boldsymbol{\phi}(\mathbf{q}, t) &= \mathbf{0} \\ \boldsymbol{\psi}(\mathbf{q}, \dot{\mathbf{q}}, t) &= \mathbf{0} \end{aligned} \quad (5.1)$$

where \mathbf{M} is the inertia matrix, $\boldsymbol{\beta}$ is the vector of momentum and momenta moments, \mathbf{q} is the generalized position vector, $\boldsymbol{\phi}$ is the system of holonomic constraint equations, $\boldsymbol{\psi}$ is the system of non-holonomic constraint equations, vectors $\boldsymbol{\lambda}_\phi$ and $\boldsymbol{\lambda}_\psi$ collect the Lagrange multipliers respectively associated with the holonomic and the non-holonomic constraints, \mathbf{f} is the vector of external loads, and (\clubsuit) represents the time derivative of (\clubsuit) .

Dynamic systems on MBDyn are based on nodes, in analogy with displacement-based Finite Element algorithms. Nodes provide degrees of freedom; they can be associated to different physical domains, such as mechanical, hydraulic, thermal and

electrical. This structure makes it easier to integrate heterogeneous components of the same machine in one simulation.

For example, the stepping motor model presented in this work integrates dynamic structural and electric nodes in one element, which simulates the interaction between mechanical and electrical domains.

Constraints, forces, voltages and currents have to be applied on, or imposed to, nodes; thus, they become part of the modeled system. This is accomplished by adding two sets of equations to variables associated with each node. The first set is used in the assembly of the system of equations (5.1), and the second set is used during the nonlinear solution phase, if it is necessary, and it requires the analytical determination of the Jacobian matrix of the Eq. 5.1.

5.2.1 Stepper Motor Formulation

The formulation of the stepper motor requires a combination of electric and structural nodes. The electric nodes receive the voltage coming from the stepper driver, which is controlled by an input signal that provides the step information (time and direction).

A stepper motor can have multiple phases; each phase has to be associated with an electric node, which has the following equation [13]:

$$\epsilon_k = V_k - R \cdot i_k - L \frac{di_k}{dt} - e_k = 0 \quad (5.2)$$

where V_k is the voltage applied by the stepper driver to the phase k , R is the resistance of the winding connected to this phase, i_k is the current in the circuit of this phase, L is the winding inductance of phase k , $\frac{di_k}{dt}$ is the time derivative of the current, and e_k is electromotive force induced.

The electromotive force e_k varies with the angular velocity of the stepper rotor:

$$e_k = K_m \cdot \sin(N_r \theta_z + \theta_{0k}) \cdot \omega_z \quad (5.3)$$

where K_m is the motor constant, N_r is the number of pole pairs, θ_z is the rotor rotation, θ_{0k} is the angular position of the k th winding in the stator, and ω_z is rotation speed of the rotor.

The electromagnetic torque is applied to the structural nodes; its magnitude is calculated according to the formula:

$$T_e = \sum_{k=1}^s K_m \cdot \sin(N_r \theta_z + \theta_{0k}) \cdot i_k \quad (5.4)$$

where T_e is the electromagnetic torque, n is the number of phases, and i_k is the phase current.

The torque generated by the motor is applied to the rotor and, therefore, there is a corresponding reaction moment in the motor stator. The structural nodes attached to the rotor and the stator are defined in absolute coordinates, thus the torque has to be transformed from local to global coordinates:

$$\begin{aligned} \mathbf{C}_1 &= -\mathbf{R}_{1r} \mathbf{k} T_e \\ \mathbf{C}_2 &= -\mathbf{C}_1 \\ \mathbf{R}_{1r} &= \mathbf{R}_1 \tilde{\mathbf{R}}_1 \end{aligned} \quad (5.5)$$

where \mathbf{C}_1 is the vector of the moment applied to the stator node, $\mathbf{k} = \{0 \ 0 \ 1\}^T$ is a unit vector pointing along direction \mathbf{Z} (third coordinate), \mathbf{C}_2 is the moment applied to the rotor node, \mathbf{R}_1 is the orientation matrix associated to the stator node, and $\tilde{\mathbf{R}}_1$ is the relative orientation matrix of the motor in relation to the stator node.

The Jacobian matrix is used by the implicit integration method employed by MBDyn. Therefore, its calculation is necessary for the analysis of the model. A perturbation can be applied to the Eq. 5.2 to determine the component to be included into the Jacobian matrix of the complete system:

$$\delta \epsilon_k = \delta V_k - R \delta i_k - L \delta \left(\frac{di_k}{dt} \right) - \delta e_k = 0 \quad (5.6)$$

The voltage V_k is defined by the stepper driver and, therefore, its value is known and its virtual value δV_k is equal to zero. The perturbation of the electromotive force is:

$$\delta e_k = K_m (N_r \cos(N_r \theta_z + \theta_{0k}) \omega_z \delta \theta_z + \sin(N_r \theta_z + \theta_{0k}) \delta \omega_z) \quad (5.7)$$

where the perturbation of the rotation speed ω_z is:

$$\begin{aligned} \delta \omega_z &= \mathbf{k}^T \delta \mathbf{R}_{1r}^T (\omega_2 - \omega_1) + \mathbf{k}^T \mathbf{R}_{1r}^T (\delta \omega_2 - \delta \omega_1) \\ &= \mathbf{k}^T \mathbf{R}_{1r}^T [(\omega_2 - \omega_1) \times] \delta \mathbf{g}_1 + \mathbf{k}^T \mathbf{R}_{1r}^T (\delta \dot{\mathbf{g}}_2 - \omega_2 \times \delta \mathbf{g}_2 - \delta \dot{\mathbf{g}}_1 + \omega_1 \times \delta \mathbf{g}_1) \end{aligned} \quad (5.8)$$

$$\begin{aligned} &= hb_0 \mathbf{k}^T \mathbf{R}_{1r}^T [(\omega_2 - \omega_1) \times] \delta \dot{\mathbf{g}}_1 \\ &+ \mathbf{k}^T \mathbf{R}_{1r}^T ((1 - hb_0 [\omega_2 \times]) \delta \dot{\mathbf{g}}_2 - (1 - hb_0 [\omega_1 \times]) \delta \dot{\mathbf{g}}_1) \end{aligned} \quad (5.9)$$

where ω_1 and ω_2 are respectively the angular velocity of the stator and the rotor nodes, \mathbf{g}_1 and \mathbf{g}_2 are respectively the vector of virtual Cayley-Gibbs-Rodriguez parameters associated to the rotation of the stator node and the rotor node, h is the time step, and b_0 is a constant that depends on the integration method [12].

The rotation angle θ_z is determined by the orientation matrices of the rotor and stator nodes:

$$\begin{aligned} \boldsymbol{\theta} &= \text{ax} \left(\exp^{-1} (\mathbf{R}_{1r}^T \mathbf{R}_2) \right) \\ \theta_z &= \mathbf{k} \cdot \boldsymbol{\theta} \end{aligned} \quad (5.10)$$

where $\boldsymbol{\theta}$ is the vector of the Euler parameters that represents the relative orientation between rotor and stator, (\clubsuit^T) represents the transposed of (\clubsuit) , ax is the inverse of the operator that generates the cross product matrix, \exp^{-1} is the operator that determines the Euler parameters from an orientation matrix, and \mathbf{R}_2 is the orientation (rotation) matrix associated to the rotor node.

The perturbation of the rotation angle $\delta\theta_z$ can be obtained through the perturbation of the orientation matrices [11]:

$$\delta\theta_z = \mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \quad (5.11)$$

$$= hb_0 \mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T (\delta\dot{\mathbf{g}}_2 - \delta\dot{\mathbf{g}}_1) \quad (5.12)$$

where $\boldsymbol{\Gamma}(\cdot)$ is the differential operator associated to the exponential map, $\boldsymbol{\theta}_{2\delta}$ is the vector of virtual rotation of the rotor node, and $\boldsymbol{\theta}_{1\delta}$ is the virtual rotation of the stator node.

The perturbation of the moment applied to the rotor node can be used to determine the perturbation of the stator node, thus:

$$\begin{aligned} \delta\mathbf{C}_1 &= \delta\mathbf{R}_{1r} [3] T_e + \mathbf{R}_{1r} [3] \delta T_e \\ \mathbf{R}_{1r} [3] &= \mathbf{R}_{1r} \mathbf{k} \end{aligned} \quad (5.13)$$

$$\delta T_e = K_m \sum_{k=1}^n (\sin(N_r \theta_z + \theta_{0k}) \delta i_k + N_r \cos(N_r \theta_z + \theta_{0k}) i_k \delta \theta_z)$$

Applying the *updated-updated* method [12], i.e. incrementally expressing rotations from their predicted value, such that the unknown rotation only accounts for the correction contribution, the virtual rotation term can be approximated by:

$$\delta\mathbf{R}_{1r} [3] T_e = -\mathbf{R}_{1r} [3] T_e \times \delta\mathbf{g}_1 \quad (5.14)$$

Using the Eq. 5.14, and replacing the virtual perturbation of the parameters with their virtual velocities, the Eq. 5.13 can be rewritten as:

$$\begin{aligned} \delta\mathbf{C}_1 &= -hb_0 [\mathbf{R}_{1r} [3] T_e \times] \delta\dot{\mathbf{g}}_1 + \mathbf{R}_{1r} [3] hb_0 K_m \sum_{k=1}^n \left(\sin(N_r \theta_z + \theta_{0k}) \delta \frac{di_k}{dt} \right. \\ &\quad \left. + N_r \cos(N_r \theta_z + \theta_{0k}) i_k \mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T (\delta\dot{\mathbf{g}}_2 - \delta\dot{\mathbf{g}}_1) \right) \\ &= -hb_0 [\mathbf{R}_{1r} [3] T_e \times] \delta\dot{\mathbf{g}}_1 + \mathbf{R}_{1r} [3] hb_0 K_m \sum_{k=1}^n \left(\sin(N_r \theta_z + \theta_{0k}) \delta \frac{di_k}{dt} \right) \\ &\quad + hb_0 K_m \sum_{k=1}^n (N_r \cos(N_r \theta_z + \theta_{0k}) i_k) \cdot \mathbf{R}_{1r} [3] \otimes \left(\mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T \right) (\delta\dot{\mathbf{g}}_2 - \delta\dot{\mathbf{g}}_1) \end{aligned} \quad (5.15)$$

The Jacobian matrix can be written using Eqs. 5.6 and 5.15:

$$\mathbf{J} = \begin{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_1} \\ \frac{\partial \mathbf{C}_2}{\partial \dot{\mathbf{g}}_1} \\ \frac{\partial \epsilon_1}{\partial \dot{\mathbf{g}}_1} \\ \vdots \\ \frac{\partial \epsilon_n}{\partial \dot{\mathbf{g}}_1} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_2} \\ \frac{\partial \mathbf{C}_2}{\partial \dot{\mathbf{g}}_2} \\ \frac{\partial \epsilon_1}{\partial \dot{\mathbf{g}}_2} \\ \vdots \\ \frac{\partial \epsilon_n}{\partial \dot{\mathbf{g}}_2} \end{bmatrix} & \begin{bmatrix} \frac{\partial \mathbf{C}_1}{\partial \frac{di_1}{dt}} \\ \frac{\partial \mathbf{C}_2}{\partial \frac{di_1}{dt}} \\ \frac{\partial \epsilon_1}{\partial \frac{di_1}{dt}} \\ \vdots \\ \frac{\partial \epsilon_n}{\partial \frac{di_1}{dt}} \end{bmatrix} & \cdots & \begin{bmatrix} \frac{\partial \mathbf{C}_1}{\partial \frac{di_n}{dt}} \\ \frac{\partial \mathbf{C}_2}{\partial \frac{di_n}{dt}} \\ \frac{\partial \epsilon_1}{\partial \frac{di_n}{dt}} \\ \vdots \\ \frac{\partial \epsilon_n}{\partial \frac{di_n}{dt}} \end{bmatrix} \end{bmatrix} \quad (5.16)$$

where the partial derivatives of the moment applied to the stator node:

$$\begin{aligned} \frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_1} &= -hb_0 K_m \sum_{k=1}^n (N_r \cos(N_r \theta_z + \theta_{0k}) i_k) \cdot \mathbf{R}_{1r} [3] \otimes (\mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T) \\ &\quad - hb_0 [\mathbf{R}_{1r} [3] T_e \times] \\ \frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_2} &= hb_0 K_m \sum_{k=1}^n (N_r \cos(N_r \theta_z + \theta_{0k}) i_k) \cdot \mathbf{R}_{1r} [3] \otimes (\mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T) \\ \frac{\partial \mathbf{C}_1}{\partial \frac{di_k}{dt}} &= \mathbf{R}_{1r} [3] hb_0 K_m \sin(N_r \theta_z + \theta_{0k}) \end{aligned}$$

And the partial derivatives of the moment applied to the rotor node:

$$\begin{aligned} \frac{\partial \mathbf{C}_2}{\partial \dot{\mathbf{g}}_1} &= -\frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_1} \\ \frac{\partial \mathbf{C}_2}{\partial \dot{\mathbf{g}}_2} &= -\frac{\partial \mathbf{C}_1}{\partial \dot{\mathbf{g}}_2} \\ \frac{\partial \mathbf{C}_2}{\partial \frac{di_k}{dt}} &= -\frac{\partial \mathbf{C}_1}{\partial \frac{di_k}{dt}} \end{aligned}$$

The partial derivatives of the circuit equation of phase k are:

$$\begin{aligned} \frac{\partial \epsilon_k}{\partial \dot{\mathbf{g}}_1} &= hb_0 K_m N_r \cos(N_r \theta_z + \theta_{0k}) \omega_z \mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T \\ &\quad - K_m \sin(N_r \theta_z + \theta_{0k}) \mathbf{k}^T \mathbf{R}_{1r}^T (hb_0 [(\omega_2 - \omega_1) \times] - (1 - hb_0 [\omega_1 \times])) \\ \frac{\partial \epsilon_k}{\partial \dot{\mathbf{g}}_2} &= -hb_0 K_m N_r \cos(N_r \theta_z + \theta_{0k}) \omega_z \mathbf{k}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1r}^T \\ &\quad + K_m \sin(N_r \theta_z + \theta_{0k}) \mathbf{k}^T \mathbf{R}_{1r}^T (1 - hb_0 [\omega_2 \times]) \\ \frac{\partial \epsilon_k}{\partial \frac{di_k}{dt}} &= -hb_0 R - L \end{aligned}$$

5.2.2 Stepper Driver Formulation

The stepper driver is responsible for translating the signal coming from a computer or a controller into voltage of each motor phase. Different waveforms can be generated by the driver, and they play an important role on the motor's behavior. Full step waveform generates the maximum rated torque of the motor and rotates a full step. If the motor has 100 steps per full rotation, this means that size of one full step is equal to 3.6° .

Microstepping waveforms are the form of getting smaller movements of motor at the cost of reducing the holding torque. It is a common strategy applied to open source 3D printers to improve their resolution, because it has small cost and it is very effective. For example, dividing a full step into 16 microsteps, it is possible to achieve a much smaller step of 0.225° using the same motor with 100 steps per full rotation.

Basically, there are two digital signals transmitted to an actual stepper driver: step and direction. The level of each one determines if the motor has to move and its direction. To mimic this behavior, the stepper driver was formulated using the following equation:

$$V_k = V_{max} \cdot \sin\left(\frac{\pi S}{c_m \cdot n} + \phi_o - \frac{k \cdot \phi_d}{n - 1}\right) \quad (5.17)$$

where V_k is the voltage applied by the stepper driver to the phase k , V_{max} is the maximum voltage applied to the motor (rated tension), S is the driver input signal, c_m is number of microsteps, n is the number of phases, ϕ_0 is the phase offset, k is the number of the phase (from 1 to n), and ϕ_d is the phase difference between phases.

The digital signals, step and direction, were translated to the variable S . To move one step forward the current value of S should be increased by one ($S_{i+1} = S_i + 1$), and to move backward it should be decrease by one ($S_{i+1} = S_i - 1$).

For instance, if S is equal to 1, and it has to move one step forward, S has to be changed to 2. To go another step forward, S has to be changed to 3. If otherwise, the motor has to move one step backward, and the current S is 2, it has to be changed to 1.

To apply the Eq. 5.17 to generate full step ($c_m = 1$) and half step ($c_m = 2$) waveforms, the following condition expression has to be applied:

$$\text{if } |V_k| > V_{threshold} \text{ then } V_k = V_{max} \frac{V_k}{|V_k|} \text{ else } V_k = 0$$

where $V_{threshold}$ is the threshold tension (e.g. $V_{threshold} = V_{max} \times 10^{-6}$).

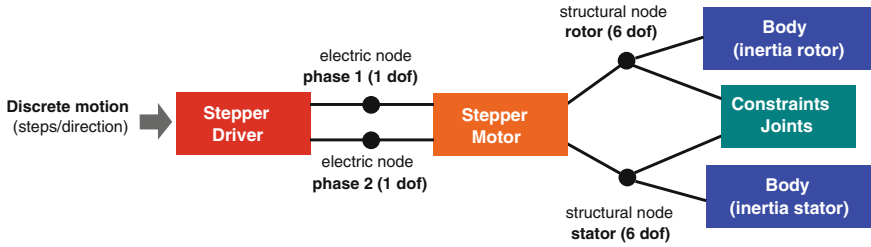


Fig. 5.3 Stepper motor system

5.2.3 Integration of the Elements of Different Domains

The mathematical model of each part has to be integrated in the system; then, the complete simulation can be run. The main entities of integration are the elements, because they provide the equations that express the interaction between domains. Figure 5.3 shows the system of one stepper motor, where the discrete motion constitutes the system input that is calculated from the inverse kinematic analysis. The stepper driver transforms the input signal into tension on the electric nodes that represent the motor phases.

The stepper motor is connected to electric and structural nodes; therefore, it relates electrical and mechanical subsystems. This is a two way connection, which is shown by Eqs. 5.2 and 5.4, where voltage and current of the electric nodes generate torque between rotor and stator, and the position of the rotor influences the current that is consumed by the motor.

The “body” elements are connected to the structural nodes of the stepper; they attribute the inertia to the rotor and the stator. These nodes are constrained by a “total joint” [11], that allows the relative rotation of the rotor inside the stator. This joint completes the modeling of the stepper motor in the MBDyn.

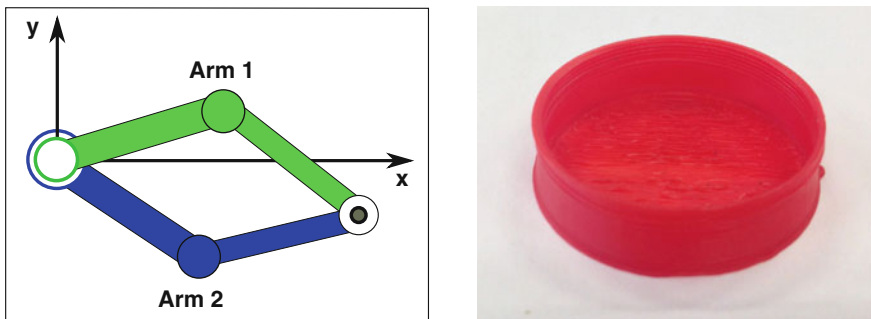


Fig. 5.4 SCARA model (left) and examples of the printed part (right)

5.3 Results

Figure 5.4 shows a diagram of the machine developed in this work. The motion of the stepper motors is transmitted to concentric shafts by a set of pulleys. The upper end of the shafts is connected to the arms of the SCARA mechanism in the origin point of the coordinate system. The ends of both arms are connected to the extrusion head, thus the combined motion of each motor makes the head to move in the xy plane. The motion of the deposition bed (table) is performed by other stepper motor, which only moves when the deposition of a new layer of the model needs to start.

The use of concentric shafts is a way to avoid a locking position during the printing process. The locking position occurs when the arm segments connected to the head form an angle of 180° .

Table 5.1 shows some parameters used in the simulation. The motor model is based on a high torque two phase hybrid stepping motor (NEMA23), that has 200 steps per turn ($=1.8^\circ$ step) and a holding torque of 90 N-cm. The pulley sets that connect each stepper motor to its respective arm have a gear ratio of 1:7.5, which

Table 5.1 Scara printer parameters

Part	Parameter	Value
Arm (one segment)	Mass	0.184 kg
	Moment of inertia I_{xx}	3.07×10^{-5} kg/m ²
	Moment of inertia I_{yy}	6.2×10^{-4} kg/m ²
	Moment of inertia I_{zz}	6.4×10^{-4} kg/m ²
	Length	200 mm
Deposition head (lighter version)	Mass	0.12 kg
	Moment of inertia I_{xx}	4.52×10^{-5} kg/m ²
	Moment of inertia I_{yy}	4.52×10^{-5} kg/m ²
	Moment of inertia I_{zz}	1.35×10^{-5} kg/m ²
Deposition head (heavier version)	Mass	0.5 kg
	Moment of inertia I_{xx}	6.21×10^{-4} kg/m ²
	Moment of inertia I_{yy}	2.05×10^{-3} kg/m ²
	Moment of inertia I_{zz}	5.21×10^{-4} kg/m ²
Stepper motor	Rotor mass	0.07 kg
	Rotor moment of inertia	5.4×10^{-6} kg/m ²
	Number of phases	2
	Winding inductance	10 mH
	Winding resistance	7.4 Ω
	Number of teeth N_r	50
	Motor constant K_m	$0.04 \cdot N_r$
	Viscous friction	0.5 kg-rad/s
	Step size	1.8 $^\circ$
Reduction (motor to arm)	Gear ratio	1:7.5

reduces the maximum speed, but improves the system precision. To make the motion even finer, the stepper driver adopts a microstepping strategy of 1/16th step, thus the motor original step is divided by 16, which yields a step of 0.1125° , and a final arm rotation of 0.015° (2.618×10^{-4} rad).

All the segments of both arms have the same length (200 mm), which is a little longer than RepRAP Morgan (150 to 170 mm) [6]. Each segment was modeled as a rigid body, that is connected to other segments by “total joint” elements that mimic revolute joints. The element “gear joint” [15] connects motors and arms, and is able to reproduce the mechanical reduction of the set of pulleys.

The extruder head was also modeled as a rigid body; two versions were simulated. The “lighter” version is based on a hot end composed by a heater and a radiator. Due to its low inertia, this model is usually adopted in smaller 3D printers, that have less powerful motors, so they can move the head with less effort. The “heavier” version is a complete extruding system [19] that includes the feeding mechanism, thus it can pull the filament directly to the hot end, which allows a much better control of the extrusion flow.

Figure 5.4 (right) shows the part chosen to have the manufacturing process simulated. It is a simple cylinder with a closed bottom. The STL model of the part was converted by the software Slic3r [18] into G-code, which is compatible to all 3D printers based on the RepRap project. The G-code was translated to the extrusion head movement by a Python script based on the Marlin firmware [10]. This algorithm limits the speed in the intersections using a trapezoid strategy, i.e., the machine accelerates the extruder until it reaches the nominal speed determined in the G-code file; when it approaches an intersection, where the motion direction usually changes, the machine needs to decelerate to the established intersection speed. This strategy reduces the error caused by sudden changes of direction, where the inertia continues to push the system into the original direction.

This script generates a trajectory for the extruder, that is used by the inverse kinematic analysis of MBDyn to convert the extruder motion into joints rotation. The resulting joint rotation is processed by a second Python script that transforms the rotation in steps and their directions, corresponding to the movement that needs to be accomplished by each motor. This script must consider the reduction between arm and motor, and the strategy defined in the stepper driver. It also considers what motor position is closest to the position required by the trajectory. The generated files are the inputs to each one of stepper motor systems used in the dynamic simulation of the SCARA printer.

The simulation of the printer produces the data regarding the dynamic behavior of all components. The final objective of this work is to verify the impact of the printer parameters on the part construction. To this end, the simulated motion of the extruding head must be compared with the planned trajectory with the help of an error function. This function can be calculated using the approach adopted in [14], where the positioning error ϵ_{pos} is the distance between the planned and the actual position of the extruding head:

$$\epsilon_{pos} = \sqrt{(x_{actual} - x_{planned})^2 + (y_{actual} - y_{planned})^2} \quad (5.18)$$

The coordinate z is not considered in the calculation, because it represents the movement of the deposition bed, which is slow and is only performed upon completion of each layer.

The distribution of the error can be determined by clustering the errors between the planned and the actual position for every time instant calculated in the simulation. The number of occurrences (frequency) is normalized by the total number of steps of the simulation.

The process of manufacturing of the chosen part takes 274.8 s, which includes the initial positioning of the head and a previous deposition, that is usually employed to verify whether the extruder is working correctly. MBDyn took about 12 min to run the simulation on Core i7 (2670). A variable time step strategy was adopted; most of the time step was 10^{-3} s. The simulation required an average of 2.8 Jacobian evaluations per step.

5.3.1 Extruder Head

In the first two simulations, the voltage supplied to the motors was set to 12 V. This voltage enables the motors to work with a higher torque; therefore, errors should be lower when compared to lower supply voltages.

Figure 5.5 shows the trajectory deviation of the printer using the lighter extruding head and a stepper supply voltage of 12 V. The biggest deviations in trajectory occur in the three highlighted areas. They are related to the accelerations or decelerations from high speed displacements, which occur when the deposition is finished in a particular area and the head has to be quickly repositioned to a new deposition area.

Figure 5.6 shows the error distribution curve (left) and the trajectory deviation in Area 1 (right) using the lighter extruding head and a stepper supply voltage of 12 V. The error distribution shows that most of positioning errors fall between 0 and $100 \mu\text{m}$ with an average of $46.1 \mu\text{m}$.

Figure 5.7 (left) shows that for the heavier head the error distribution is very similar to the distribution presented in Fig. 5.6, which means that the motor torque is enough to hold the increase of the head inertia. There is a slight change in the average error ($47.4 \mu\text{m}$), but considering that the head mass increased four times, it is worth changing the extruder with a heavier version and gain more control over the deposition flow. Figure 5.7 (right) confirms that increasing the mass had little influence on the positioning error.

Figure 5.8 compares the performance of both extruders in the areas denoted by Fig. 5.5. Although the trajectories are not identical, the deviation amplitudes are quite similar. The heavier model seems to take longer to reach the position determined by

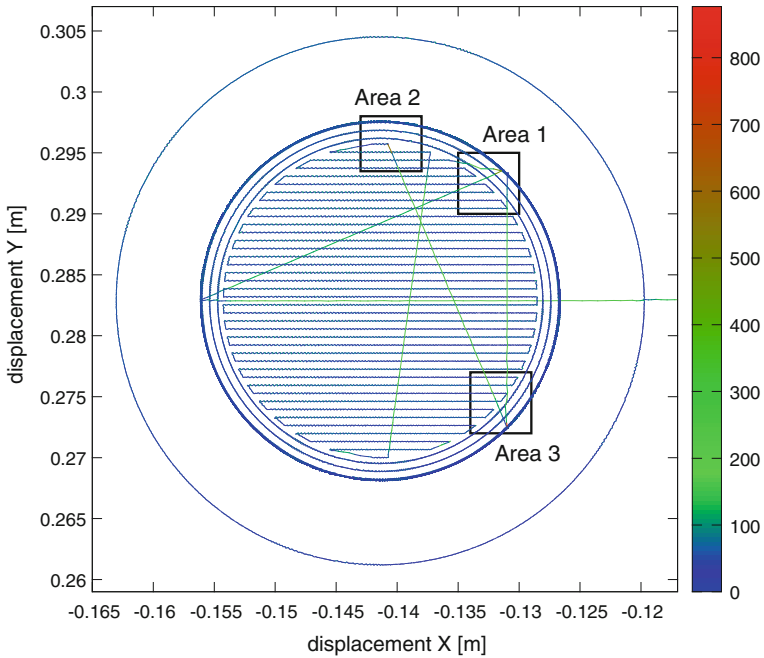


Fig. 5.5 Trajectory deviation of the machine with the lighter head (0.12 kg) and a stepper supply voltage of 12 V

the stepper motor. Another problem is that the motion of the stepper motor is discrete (by steps), thus the trajectory does not always coincide with the motor rotation. This problem is mitigated by the transmission reduction and microstepping, but not eliminated.

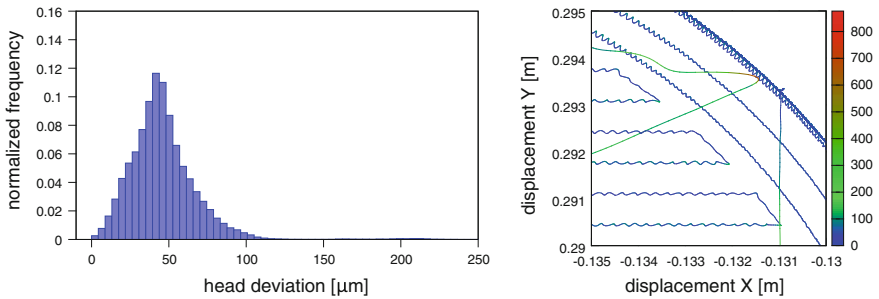


Fig. 5.6 Error distribution (*left*) and trajectory deviation in Area 1 (*right*) of the machine with the lighter head and supply voltage of 12 V

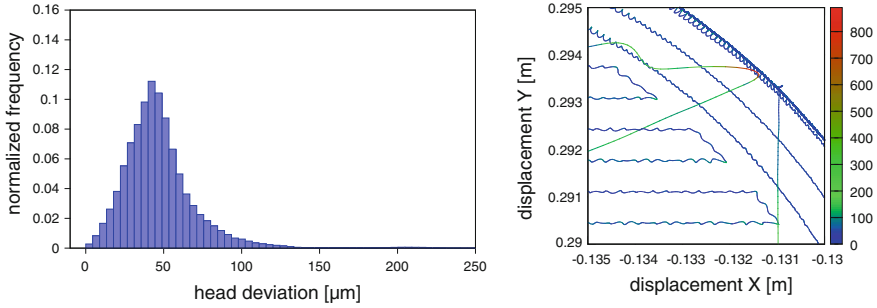


Fig. 5.7 Error distribution (*left*) and trajectory deviation in Area 1 (*right*) of the machine with the heavier head (0.5kg) and a stepper supply voltage of 12 V

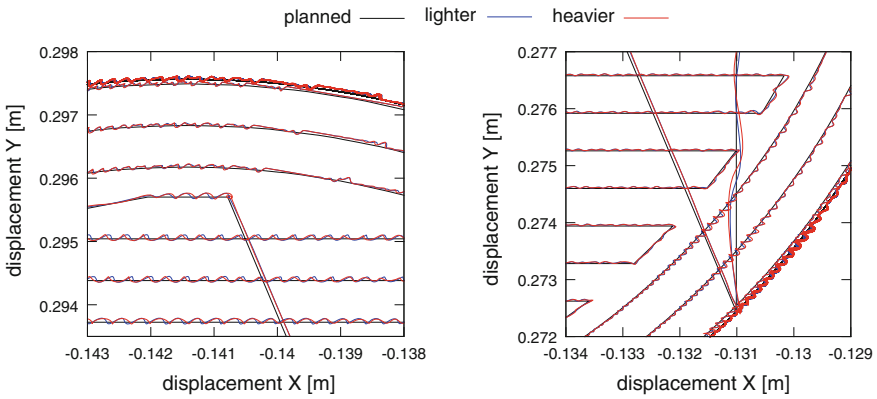


Fig. 5.8 Comparison between the lighter and the heavier extruder (*left* Area 2; *right* Area 3)

5.3.2 Stepper Voltage Supply

The results show that the voltage supply is enough to overcome an increase of the extruding head, therefore, it is interesting to verify what happens if the voltage supplied by the stepper driver is decreased. The two simulations of this section uses 5 V instead of 12 V, with the same head models.

The distribution of the error shows a higher frequency in the range of 30–60 μm in Fig. 5.9 (left), when it is compared to the simulations using 12 V. The average error is also higher (48.2 μm), which reveals that less voltage is translated to less torque to control the motion of the extruder.

Figure 5.10 (left) shows that the increase of the mass takes to larger errors, which is indicated by the increase of the frequency in the range from 100 to 220 μm. The biggest errors increased from 0.8 to 1.2 mm in the same transition points of the previous simulations. The influence of the heavier extruder in the trajectory deviation can be noticed when comparing Figs. 5.9 (right) and 5.10 (right). This can

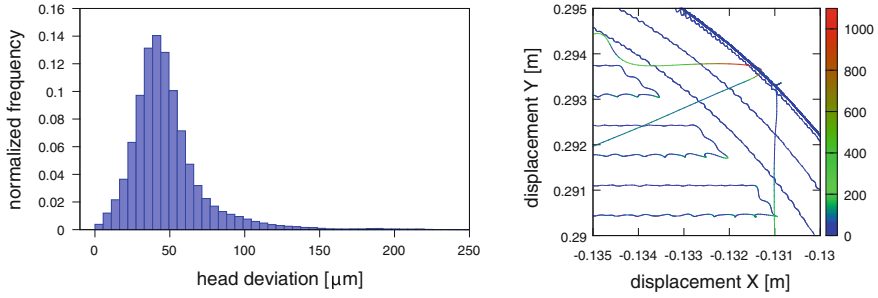


Fig. 5.9 Error distribution (*left*) and trajectory deviation in Area 1 (*right*) of the machine with the lighter head (0.12 kg) and a stepper supply voltage of 5 V

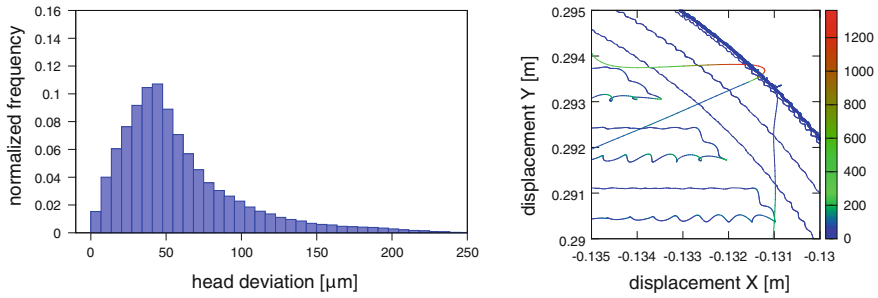


Fig. 5.10 Error distribution (*left*) and trajectory deviation in Area 1 (*right*) of the machine with the heavier head (0.5 kg) and a stepper supply voltage of 5 V

be explained by the fact that inertia plays a greater influence on the head movement, and the stepper motors cannot provide the necessary torque to control the movement.

Figure 5.11 shows that the deviation of the heavier extruder has a larger amplitude than that of the lighter one. The detail of Area 3 (Fig. 5.11—left) shows that the error increases when there is a change of the motion direction, which is visible when the head is moving to the next filling line.

5.3.3 Microstepping

Adopting smaller microsteps is a form to get higher resolution, however, less holding torque is available to control the mechanism position. To verify if it has some effect on the model of the SCARA printer, the strategy of microstepping was changed decreasing the microstep to 1/32th of a step, i.e. each step is half of the size of the simulations presented in the former sections. To highlight the effect of torque loss the supply voltage was kept at 5 V.

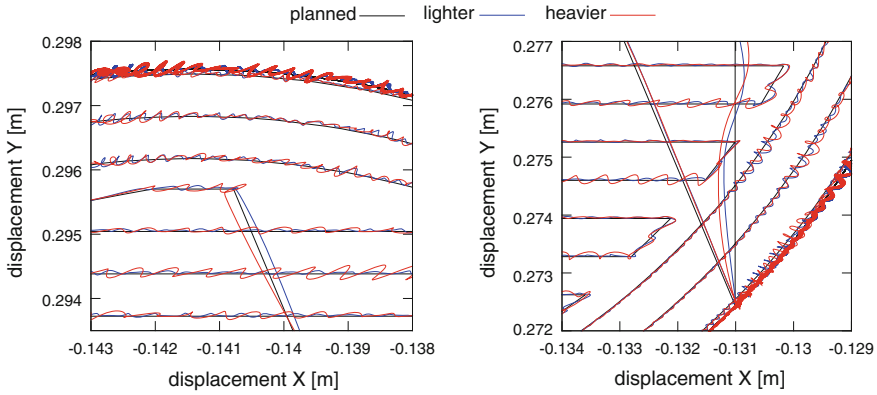


Fig. 5.11 Comparison between the lighter and the heavier extruder (*left* Area 2; *right* Area 3)

Figure 5.12 shows the distribution of the error using smaller microsteps, and it can be noticed that the higher frequency is now in the range of 0–50 μm , as opposed to 0–100 μm gotten of a microstep size of 1/16th. The shape of the distribution changed from Figs. 5.9 and 5.10 indicating that the effect of microstepping is not linear, although most of the errors fall in half of the former range.

The maximum errors were approximately 1.1 mm to the lighter head and 1.3 mm to the heavier head, which indicates a small increase in the error in the movement transitions that can be related to the torque loss of this new microstepping strategy.

Figure 5.13 shows that except in the transition point, the trajectory of both extruders is closer to the planned one. The top section of Area 1 and the right inferior section of Area 2 (Fig. 5.13) highlight the difference in the construction of the cylinder wall when comparing them to Fig. 5.13.

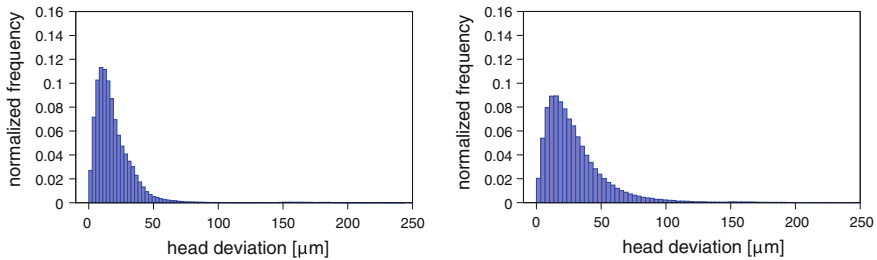


Fig. 5.12 Error distribution of the machine with the lighter head (*left*) and the heavier head (*right*) using a 1/32th step

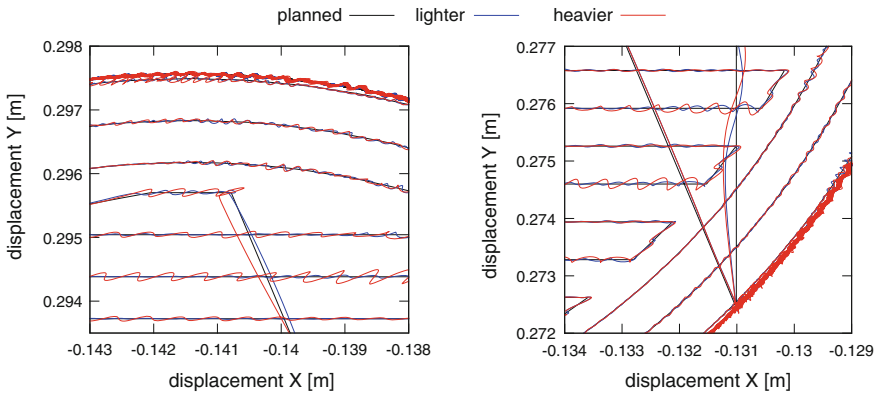


Fig. 5.13 Comparison between the lighter and the heavier extruder (*left* Area 2; *right* Area 3)

5.3.4 Results Summary

Table 5.2 shows the trajectory deviation for different voltages, head masses and microstepping strategies. The combination of low voltage, heavier extruder and larger microstep showed the worst performance of all simulations. The sample standard deviation of the error in this case is twice the best configuration (lighter head and smaller microstep).

It can also be noticed that increasing the motor torque by using a higher voltage can be a good solution to compensate heavier extruders. However, the best solution to improve the printer resolution, in this case, was to decrease the microstep size to 1/32th.

Table 5.3 shows the power supplied by the driver to the stepper motor. The power supplied when voltage is set to 12 V is five times higher than the 5 V supply, which means that a more powerful driver is needed when the system operates at higher voltages. Open source printers usually adopt stepper drivers (e.g. A4988) that can deal with an average current of 1 A with peaks of 2 A, which is less than it is required

Table 5.2 Trajectory deviation

Supply voltage (V)	Head mass (kg)	Microstep size	Extruding head deviation (μm)	
			Mean	Standard deviation
5	0.12	1/16	48.2	26.5
5	0.5	1/16	58.7	42.2
12	0.12	1/16	46.1	22.7
12	0.5	1/16	47.4	24.5
5	0.12	1/32	19.7	20.2
5	0.5	1/32	30.3	27.8

Table 5.3 Power supplied by the stepper driver

Supply voltage (V)	Head mass (kg)	Microstep size	Motor 1 (W)		Motor 2 (W)	
			Mean	Maximum	Mean	Maximum
5	0.12	1/16	3.41	13.03	3.39	13.08
5	0.5	1/16	3.41	13.34	3.40	13.11
12	0.12	1/16	19.52	41.07	19.49	43.55
12	0.5	1/16	19.52	40.55	19.49	44.06
5	0.12	1/32	3.41	13.06	3.40	13.15
5	0.5	1/32	3.41	13.38	3.40	13.14

by the lighter configuration (13 W \rightarrow 2.6 A). In this case, the speed has to be limited, otherwise, the overcurrent protection system would cut the motor tension.

Adopting a smaller microstep did not take to a significant increase of the power supplied by the stepper driver, which indicates that a driver with a strategy with smaller steps is better than a more powerful one in this case.

5.4 Conclusions

The modeling of the 3D printer based on a SCARA mechanism showed that it is possible to use a multidisciplinary platform to simulate a coupled problem. Besides that, MBDyn was also used to generate the control sequence of each stepping motor through the inverse kinematic analysis.

To simulate the printer motors, a mathematical model of a hybrid stepping motor was implemented inside MBDyn, along with a stepper driver to control the motor motion. Although it is not easy to determine the analytical Jacobian matrix of these mathematical models, the advantage of having a smooth and relatively fast simulation overcomes the problems to implement it.

Electrical and mechanical components are usually hard to integrate, especially when the model incorporates digital components, such as the motor driver. Sudden changes of the input signal causes high frequencies that the differential solvers do not manage easily. The combination of the nonlinear solver of the implicit solver and a variable time step seemed to solve the problems of mixing components of different domains that operate at different rates.

The 3D printing process adds a level of difficulty to the simulation, because the modeling of the part to be built and its translation to machine movements require a previous knowledge of the equipment and material to be printed, and they change the way this translation is made.

The simulation of dynamic behavior of the printer helps to understand what can be changed to the design of this machine. For instance, the effect of increasing the

mass of the extruder head can be overcome by increasing the stepping motor supply. However, the driver must be capable of providing enough power to feed the stepper, otherwise, the driver protection system cuts the current introducing a perturbation in the motion.

Another conclusion of this work is that adopting a stepper driver that works with smaller microsteps can be more effective to improve the printer resolution than increasing the supply voltage.

Acknowledgments This work was supported by CNPq (National Council for Scientific and Technological Development—Brazil).

References

1. Al Mashhadany YI (2012) SCARA robot: modeled, simulated, and virtual-reality verified. In: Ponnambalam SG et al (eds) Trends in intelligent robotics, automation, and manufacturing. Springer, Berlin, pp 94–102
2. Amiri-M AA, Gharib MR, Moavenian M, Torabiz K (2009) Modelling and control of a SCARA robot using quantitative feedback theory. *Proc Inst Mech Eng I J Syst Control Eng* 223(7): 919–928
3. Andeen G, Kombluh R (1988) Design of compliance in robots. In: IEEE (ed) Proceedings of the IEEE international conference on robotics and automation
4. Das MT, Dülger LC (2005) Mathematical modelling, simulation and experimental verification of a scara robot. *Simul Model Pract Theory* 13(3):257–271
5. Fumio O (1993) Evaluation of a rigid body model for a SCARA-type robot with harmonic drive transmissions. In: IEEE (ed) Proceedings of the 1993 IEEE/RSJ international conference on intelligent robots and systems, vol 2, pp 787–794
6. Harley Q (2013) RepRap Morgan. http://reprap.org/wiki/RepRap_Morgan. Accessed 15 Apr 2015
7. Hermle M, Eberhard P (2000) Control and parameter optimization of flexible robots. *Mech Struct Mach* 28(2–3):137–168
8. Joubair A, Slamani M, Bonev IA (2013) Kinematic calibration of a five-bar planar parallel robot using all working modes. *Robot Comput-Integr Manuf* 29(4):15–25
9. Kalra P, Mahapatra PB, Aggarwal DK (2006) An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. *Mech Mach Theory* 41(10):1213–1229
10. Lahteine S (2015) Marlin 3D Printer Firmware. <http://github.com/MarlinFirmware/Marlin>. Accessed 15 Apr 2015
11. Masarati P (2013) A formulation of kinematic constraints imposed by kinematic pairs using relative pose in vector form. *Multibody Syst Dyn*. doi:10.1007/s11044-012-9320-0
12. Masarati P, Morandini M, Mantegazza P (2014) An efficient formulation for general-purpose multibody/multiphysics analysis. *ASME J Comput Nonlinear Dyn* 9(4):041001. doi:10.1115/1.4025628
13. Morar A (2003) Stepper motor model for dynamic simulation. *Acta Electroteh* 44(2):117–122
14. Okabe E (2011) Dynamic modeling of a deposition mechanism of a freeform manufacturing machine. In: ABCM (ed) Proceedings of the 21st Brazilian congress of mechanical engineering—COBEM2011
15. Okabe E, Masarati P (2014) Detailed modeling of wind turbine gear set by general-purpose multibody dynamics. In: ASME (ed) Proceedings of the international design engineering technical conferences—IDETC2014, DETC2014/MECH-34898
16. Padhy SK (1992) On the dynamics of SCARA robot. *Robot Auton Syst* 10(1):71–78

17. Pigani L, Gallina P (2014) Cable-direct-driven-robot (CDDR) with a 3-link passive serial support. *Robot Comput-Integr Manuf* 30(3):265–276
18. Ranellucci A (2015) Slic3r: G-code generator for 3D printers. <http://slic3r.org>. Accessed 15 Apr 2015
19. RepRap Project (2015) Category: Extruders. <http://reprap.org/wiki/Extruder>. Accessed 15 Apr 2015
20. Seward N (2014) Wally Printer. <http://reprap.org/wiki/Wally>. Accessed 15 Apr 2015
21. Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) *Robotics: modelling, planning and control*. Springer, Berlin
22. Spong M, Hutchinson S, Vidyasagar M (2005) *Robot modeling and control, symbolic modeling of multibody systems*. Wiley, New York
23. Sprenger B, Kucera L, Mourad S (1998) Balancing of an inverted pendulum with a SCARA robot. *IEEE/ASME Trans Mechatron* 3(2):91–97
24. Tiankov T, Genova P, Kotev V, Kostadinov K (2009) Strategy for control of a hybrid macro-micro robot with a 5-link closed structure - an inverse problem of kinematics. In: *Proceedings of the international conference on manufacturing systems - ICMaS*
25. Van Helvoort J, Steinbuch M, Lambrechts P, Van de Molengraft R (2004) Analytical and experimental modelling for gain scheduling of a double scara robot. In: *Proceedings of the IFAC symposium on mechatronic systems*
26. Visioli A, Legnani G (2002) On the trajectory tracking control of industrial SCARA robot manipulators. *IEEE Trans Ind Electron* 49(1):224–232

Chapter 6

Structure Preserving Optimal Control of a Three-Dimensional Upright Gait

Michael W. Koch and Sigrid Leyendecker

Abstract The optimal control of human locomotion requires simulation techniques, which handle the contact's establishing and releasing between foot and ground. In this work, our aim is to optimally control the human upright gait using a structure preserving variational integrator, whereby different physiologically motivated cost functions are chosen and the obtained results are analysed with regard to the gait of humans. Thereby, the implemented three-dimensional rigid multibody system enables us to model forefoot as well as heel contact. The contacts between feet and ground are modelled as perfectly plastic impact and the orientation of the contact forces prevent penetration of the ground. To guarantee the structure preservation and the geometrical correctness, the non-smooth problem is solved including the contact configuration, time and force, in contrast to relying on a smooth approximation of the contact problem via a penalty potential. The applied mechanical integrator is based on a discrete constrained version of Lagrange-d'Alembert principle, which yields a symplectic momentum preserving method (see Leyendecker et al., *Optim Control Appl Methods* 31:505–528, 2009, [31] for details).

6.1 Introduction

The human environment consists of a large variety of mechanical and biomechanical systems, in which different types of contact can occur. The occurring adhesion and friction are essential to guarantee the functionality. A lot of mechanical safety systems utilise adhesion like the falling protection of lifts or the climbing security equipment. But friction is also necessary to enable the function of mechanical systems, like belt drives and braking systems. Biomechanical literature is often focussed on the function

M.W. Koch · S. Leyendecker (✉)
Chair of Applied Dynamics, University of Erlangen-Nuremberg,
91058 Erlangen, Germany
e-mail: sigrid.leyendecker@fau.de

M.W. Koch
e-mail: michael.koch@fau.de

and structure of the human locomotor system in combination with the foot-ground contact, whereby cyclic walking movements come to the fore [14, 46]. Here, we are interested in the upright bipedal gait. In contrast to movements with rolling wheels or crawler drives, the simulation of human locomotion requires the knowledge of how the contact establishing and releasing between forefoot respectively heel and ground works and how the motion sequence changes due to the contact conditions. The investigated contact formulation covers the theory of perfectly plastic contacts (e.g. see [24]), which means that the foot stays in contact with the ground for a certain time.

The straight posture of the homo sapiens' gait is a characteristic attribute of the human species, whereby the evolution of the human gait results from an anthropological optimisation process. Initiated by climbing down from trees and leaving forests, the survival in velds necessitates a physical adaptation. The results are reflected by the kind of human motion and by the physique: An upright gait shows benefits such as a distinct all-round visibility, it reduces the water loss as a consequence of evaporation and allows the possibility to use tools and weapons in the struggle to survive (further details are specified in [51]). In palaeoanthropology, the upright gait is appreciated as a key event of the human evolution with great changes of the anatomy: The human skeleton is optimised for bipedal walking with the result of an efficient and economical locomotion. In the human environment, the possibility to walk on different terrains is used to overcome different levels by stairs or ladders as noticed in [16].

In biomechanical literature, the focus is often on analysing and simulating the upright gait in conjunction with muscle models for the actuation like in [3, 21, 25, 34], whereas in robotics the aim is to develop bipedal robots with the humanlike capability to move in various circumstances (compare [11, 14, 15, 35]). In the area of computer graphics, a specific challenge is to create realistic movements for the virtual characters in video games or movies. According to the different research interests, a large variety of models exists to study bipedal walking. The models in the field of biomechanics range from a simple spring mass systems [23]—explaining the basic dynamics of walking and running—to complex biomechanical multibody systems with corresponding muscle models in [13]. As described in [16], these models are primarily used to investigate the acting forces within the body, but not to generate motions. In computer graphics, a lot of research is addressed to synthesise plausible motions. Another further interesting point is that in [44, 47] optimisation techniques are used to find a transition between two pre-recorded motions to obtain a complete movement sequence (see [16]).

A large variety of biomechanical as well as robotics literature is concerned with the human locomotor system. This has several reasons, such as a certain fascination with a model and its simulation and analysis of the human locomotor system and transformation of humanlike properties on technical systems like humanoid robots. An important result of the upright gait is that the upper extremities have no supporting

functions during the locomotion. Consequently, these extremities undertake the task of a complex and versatile gripper tool. Humanoid robots have become very popular in research and it is no longer the primary goal to reproduce only the human movement and grasping apparatus, but rather modelling of an emotional humanoid body language (see [17, 26, 39]). Another aspect is that robots can undertake testing tasks as realised e.g. for military protective clothing by the humanoid PETMAN and illustrated on [10]. This humanoid robot reproduces the human movement capability of the torso, the lower and upper extremities. It furthermore simulates human body temperature and sweating. Target of all these works is to reproduce the emotional and physiological characteristics of a real human.

In robotics, the geometry and the actuation of locomotor systems are crucial for efficient and enduring movements. From the comprehension of human walking motion as well as other humanlike movements, new ideas and approaches arise, which can be transferred in the fields of robotics, for high-tech prostheses or orthopaedic devices. Hereby several usages are thinkable like the support of nursing staff or personalised rehabilitation exercises for patients with movement restrictions. Human gait, jumping and running motions or rising from a chair are exemplarily examined in [1, 21, 22, 25, 34, 41, 42]. One aim of all these cited references is to actuate the multibody system by adequate muscles models—often 1-dimensional Hill-based muscle models are applied. These models yield only the magnitude of the acting muscular force. Consequently, the real existing three-dimensional complex anatomy, i.e. the distribution of muscle fibres and the interaction with other muscles, bones and human tissue, can only be reproduced rudimentarily by these Hill-type muscles. Occasionally, this simple 1-dimensional kind of actuation is scoffingly illustrated as ‘marionette-actuation’, see [45].

Despite all these considerations, the attention in this work in context of the walking movements is focused on the contact formulation between foot and ground as well as on the optimised motion sequence itself—this concerns the transition from a double contact to a single contact phase and vice versa. Particularly important in connection with the contact modelling is the prevention of an artificial fixing of the contact point at the ground, which involves the correct orientation of the contact forces and the adhesion forces with strict observance of Coulomb’s static friction law. As a result of these restrictions, the contact modelling has a dominant role in the simulated human movements. In an analogous manner as done for the monopedal jumper model in [27], the walker model is only actuated in the hip, knee and ankle joints by torques, which means that no external actuation is planned.

Walking differs from running by the feet’s contact sequence: Walking is characterised by switching between single and double support phases while running movements alternate between single support and flight phase. In simpler walking models, the double support phase is often ignored, but herein it is an essential part of the motion sequence. Another aspect concerns the three different contact possibilities of each foot during the walking motion: namely, we differentiate between fore-foot, heel and double contact. Finally, the optimisation of humanlike bipedal walking requires the correct consideration of the single and double support phases and also the

different contact scenarios for each foot. The reduction to an armless bipedal walker represents the minimal multibody system to simulate humanlike upright gait. Hence, the walker consists of an upper body representing the human torso and each leg consists of three rigid bodies, which represent thigh, calf and the foot. The inclusion of the feet leads to movements that differ from those considered e.g. in [16] with point feet.

The aim of this work is to compute physically valid and humanlike walking movements by using the DMOCC approach, which means that the dynamics of the multibody system is discretised by the variational integrator introduced in [30–32, 36] and the resulting discrete equations of motion serve as equality constraints for the optimisation problem in which different physiologically motivated cost functions are tested. In contrast to forward and inverse dynamic simulations, neither the trajectory nor the control sequence has to be exactly known for the walking system, because an optimal trajectory and the corresponding actuation is determined by the optimisation process with regard to the specific objective function. The optimally controlled walker allows actuation in the hip and the knee joints as well as in the two ankle joints, such that a physiologically motivated cost function is minimised. In the numerical solution, a direct transcription method is used to transform the optimal control problem into an optimisation problem being constrained by the fulfilment of discrete equations of motion, boundary conditions and path constraints, see e.g. [28, 53]. The walking motions are subdivided in two phases and they are called single and double support phases. To avoid an artificial restriction of the optimisation problem's phases by prescribing the time of contact establishing or releasing, variable time steps are used, wherefore the necessary scaling parameters are part of the optimisation parameters.

Section 6.2 describes briefly the multibody formulation in redundant coordinates and introduces a corresponding actuation force formulation. The symplectic momentum consistent variational integrator and the null space method with nodal reparametrisation, which reduces the numerical effort, are introduced in Sect. 6.3. Section 6.4 covers the optimal control problem and explains shortly the transformation into a finite dimensional optimisation problem. The bipedal walker model is described in Sect. 6.5 and the discrete equations of motion corresponding to the perfectly plastic contact in the variational approach are given. In Sect. 6.6, the general human gait sequence is described and in Sect. 6.7, the discrete constrained optimisation problem is formulated. The results for the bipedal walking with respect to four different physiological cost functions are presented before concluding the paper.

6.2 Rigid Multibody Configuration and Actuation

In this work, a rotation free formulation introduced in [7] for rigid bodies and in [6] for rigid multibody systems is used to describe the time-dependant configuration and to simulate the dynamics. The α th rigid body is specified by a configuration vector $\mathbf{q}^\alpha(t) \in \mathbb{R}^{12}$ composed by the placement of its center of mass $\boldsymbol{\varphi}^\alpha(t)$ and the

body-fixed right-handed director triad $\mathbf{d}_i^\alpha(t)$ for $i = 1, 2, 3$. This director triad specifies the body's orientation in space and has to stay orthonormal during the motion in the considered time interval $[t_0, t_N]$, which is guaranteed by six so-called internal constraints for each body, summarised in $\mathbf{g}_{int}(\mathbf{q}) = \mathbf{0} \in \mathbb{R}^{m_{int}}$.

In multibody systems, the rigid bodies are interconnected by different types of joints, e.g. revolute or spherical joints, and the resulting constraints are summarised in an external constrained vector $\mathbf{g}_{ext}(\mathbf{q}) = \mathbf{0} \in \mathbb{R}^{m_{ext}}$. The interconnections of rigid bodies as well as their rigidity gives rise to a scleronomic and holonomic constraint function $\mathbf{g}(\mathbf{q}) = [\mathbf{g}_{int}(\mathbf{q}) \ \mathbf{g}_{ext}(\mathbf{q})]^T \in \mathbb{R}^m$ on the redundant configuration variable $\mathbf{q} \in \mathbb{R}^k$, where k equals 12 times the number of bodies. The multibody systems are actuated directly by the independent generalised forces and torques $\boldsymbol{\tau} \in \mathbb{R}^{k-m}$ and the resulting k -dimensional redundant actuation $\mathbf{f}(\mathbf{q}) \in \mathbb{R}^k$ can be computed via $\mathbf{f}(\mathbf{q}) = \mathbf{B}^T(\mathbf{q}) \cdot \boldsymbol{\tau}$ with the input transformation matrix $\mathbf{B}^T(\mathbf{q}) \in \mathbb{R}^{k \times (k-m)}$. Note that the transformation matrix depends on the rigid bodies' interconnection and it is described in detail in [31].

6.3 Structure Preserving Integration for Constrained Mechanical Systems

The dynamics of time-continuous mechanical systems can be described using the Lagrangian or Hamiltonian formalism—in this work, discrete Lagrangian mechanics is used to derive a structure preserving integrator, see e.g. [38], and the constrained mechanical system is considered in a configuration manifold $\mathcal{Q} \subseteq \mathbb{R}^k$ with the time-dependent configuration vector $\mathbf{q}(t) \in \mathcal{Q}$. Corresponding to the approach in [31], the constrained version of the Lagrange-d'Alembert principle is discretised using the time nodes $\{t_0, t_1 = t_0 + \Delta t, \dots, t_n = t_0 + n\Delta t, \dots, t_N = t_0 + N\Delta t\}$, where $N \in \mathbb{N}$ is the number of time intervals and the discrete configurations $\mathbf{q}_n \approx \mathbf{q}(t_n)$ approximate the continuous trajectory. Similarly, $\boldsymbol{\lambda}_n \approx \boldsymbol{\lambda}(t_n)$ approximates the Lagrange multipliers $\boldsymbol{\lambda}(t) \in \mathbb{R}^m$. As usual in the context of discrete variational mechanics, the discrete Lagrangian $\mathcal{L}_d : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathbb{R}$ approximates the action integral of the continuous Lagrangian. Finally, the discrete Lagrange-d'Alembert principle requires stationarity of the resulting augmented action sum, i.e.

$$\begin{aligned} \delta S_d = & \delta \sum_{n=0}^{N-1} \left[\mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) - \frac{1}{2}(t_{n+1} - t_n) (\mathbf{g}^T(\mathbf{q}_n) \cdot \boldsymbol{\lambda}_n - \mathbf{g}^T(\mathbf{q}_{n+1}) \cdot \boldsymbol{\lambda}_{n+1}) \right] \\ & + \sum_{n=0}^{N-1} [\mathbf{f}_n^- \cdot \delta \mathbf{q}_n + \mathbf{f}_n^+ \cdot \delta \mathbf{q}_{n+1}] = 0 \end{aligned}$$

for all variations $\{\delta \mathbf{q}_n\}_{n=0}^N$ and $\{\delta \boldsymbol{\lambda}_n\}_{n=0}^N$ with $\delta \mathbf{q}_0 = \delta \mathbf{q}_N = \mathbf{0}$. This leads to the $(k + m)$ -dimensional constrained forced discrete Euler-Lagrange equations

$$D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) - \mathbf{G}_d^T(\mathbf{q}_n) \cdot \boldsymbol{\lambda}_n + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^- = \mathbf{0} \quad (6.1)$$

$$\mathbf{g}(\mathbf{q}_{n+1}) = \mathbf{0}, \quad (6.2)$$

for $n = 1, \dots, N-1$. Here $\mathbf{G}_d = \frac{1}{2}(t_{n+1} - t_{n-1}) \frac{\partial \mathbf{g}(\mathbf{q}_n)}{\partial \mathbf{q}}$ denotes the $(m \times k)$ -dimensional Jacobian matrix of the constraints, and $\mathbf{f}_n^- = \frac{1}{2}(t_{n+1} - t_n) \mathbf{B}^T(\mathbf{q}_n) \cdot \boldsymbol{\tau}_n$, respectively $\mathbf{f}_{n-1}^+ = \frac{1}{2}(t_n - t_{n-1}) \mathbf{B}^T(\mathbf{q}_n) \cdot \boldsymbol{\tau}_{n-1}$ are called discrete redundant actuation forces. In [38] it is shown that due to the variational derivation, the discrete trajectory inherits certain characteristic geometric properties (see e.g. [4, 37, 40]), from the continuous system and the numerical integration is thus called structure preserving. In particular, the discrete trajectories preserve a discrete symplectic form and are consistent in momentum maps (discrete Noether theorem). The latter means that in the presence of symmetry in the discrete Lagrangian, the corresponding momentum maps (e.g. linear or angular momentum) are conserved exactly along the discrete trajectory. In presence of non-symmetrical potentials like gravity or of external forces, the momentum map changes only and exactly according to the gravitational or to the external forces. Due to these structure preservation properties, the integrator is called symplectic momentum scheme. A further benefit of this mechanical integrator is the good energy behaviour, this means that there is no numerical gaining or dissipation of energy.

According to [5, 6], we apply the discrete null space method to reduce the dimension of the constrained forced discrete Euler–Lagrange equations (6.1) and (6.2). The discrete null space matrix $\mathbf{P}(\mathbf{q}_n) \in \mathbb{R}^{k \times (k-m)}$ fulfils the property $\mathbf{G}_d(\mathbf{q}_n) \cdot \mathbf{P}(\mathbf{q}_n) = \mathbf{0}$ and premultiplying Eq. (6.1) by the transposed null space matrix, the constraint forces and thereby the Lagrange multipliers vanish. The resulting k -dimensional system is called reduced forced discrete Euler–Lagrange equations. In a further step, the minimal dimension of the system can be achieved using the vector of incremental generalised coordinates $\mathbf{u}_{n+1} \in U \subseteq R^{(k-m)}$ to reparametrise the configuration vector \mathbf{q}_{n+1} in the neighbourhood of \mathbf{q}_n . The nodal reparametrisation function $\mathbf{F}_d : U \rightarrow Q$

$$\mathbf{q}_{n+1} = \mathbf{F}_d(\mathbf{u}_{n+1}, \mathbf{q}_n)$$

fulfils the constraint conditions and therefore Eq. (6.2) becomes unnecessary. Finally, the number of unknowns and thereby the numerical effort is reduced by the formulation in discrete generalised coordinates $\mathbf{u}_d = \{\mathbf{u}_n\}_{n=0}^N$ and discrete generalised forces and torques $\boldsymbol{\tau}_d = \{\boldsymbol{\tau}_n\}_{n=0}^{N-1}$. The dimension of the discrete equations of motion is reduced to $k - m$.

$$\begin{aligned} \mathbf{P}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{F}_d(\mathbf{u}_{n+1}, \mathbf{q}_n)) \\ + \frac{1}{2} \mathbf{B}^T(\mathbf{q}_n) \cdot [(t_n - t_{n-1}) \boldsymbol{\tau}_{n-1} + (t_{n+1} - t_n) \boldsymbol{\tau}_n]] = \mathbf{0} \end{aligned} \quad (6.3)$$

6.4 Optimal Control Problem

The goal of an optimal control problem is to determine the optimal state trajectory and force field for a holonomically constrained system, which moves from the initial state $\mathbf{q}(t_0) = \mathbf{q}^0, \dot{\mathbf{q}}(t_0) = \dot{\mathbf{q}}^0$ to a final state $\mathbf{q}(t_N) = \mathbf{q}^N, \dot{\mathbf{q}}(t_N) = \dot{\mathbf{q}}^N$. The investigated system fulfils the equations of motion and at the same time the objective functional

$$J(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}) = \int_{t_0}^{t_N} C(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}) dt$$

is minimised, where the integrand $C(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}) : TQ \times T_q^*Q \rightarrow \mathbb{R}$ is a given cost function. The optimal control problem is solved using a direct transcription method, which transforms it into a constrained optimisation problem. The discrete objective function approximates the integral of the continuous cost function and the discrete constrained optimisation problems reads

$$\min_{\mathbf{u}_d, \boldsymbol{\tau}_d} J_d(\mathbf{u}_d, \boldsymbol{\tau}_d) = \min_{\mathbf{u}_d, \boldsymbol{\tau}_d} \sum_{n=0}^{N-1} C_d(\mathbf{u}_n, \mathbf{u}_{n+1}, \boldsymbol{\tau}_n)$$

subject to the constraints given by the reduced discrete equations of motion of the symplectic momentum scheme in Eq. (6.3). In addition to the discrete equations of motion of the specific mechanical integrator, further constraints, like initial conditions, final conditions or periodic boundary conditions and possible equality and inequality path constraints—specified in detail in Sect. 6.7—can be imposed.

6.5 Humanoid Bipedal Walker Model

The simulation of humanoid walking requires an adequate reproduction of the human locomotor system, whereby especially movement abilities at the hip, knee and ankle joint are absolutely necessary to generate a natural gait. It is clear that pelvis, vertebrae as well as swinging arms play an important role in human walking [12, 33, 43]. The swinging arms of the opposite side of the body in respect to the swinging lower limbs reduce the angular momentum of the human body and enable a better balancing of the rotational motion caused by the motion of the legs. However, as a simplification in this work, their dynamics are summarised in a torso represented by a rigid body as done before e.g. for the monopodal jumper model in [27] or as for the bipedal walker model in [3]. The aim of this model reduction is to strictly investigate the dynamics of the lower extremities with respect to different cost functions in conjunction with a realistic contact establishing and releasing sequence. Therefore, we reduce our model to a minimal level of detail which enables us to effectively investigate and evaluate the described research goals.

The human upright gait is analysed by using an elementary model consisting of seven rigid bodies, whereby each of the legs is modelled by three rigid bodies as illustrated in Fig. 6.1. The shape of the bodies is characterised by the inertia parameters given Table 6.1. The right and left leg are absolutely identically and the hip and ankle are modelled as a spherical joint. Thigh and calf are connected via revolute joints, where the unit vectors \mathbf{n}^2 and \mathbf{n}^5 in the right and left thigh represent the axis of rotation. Consequently, the locomotor system consists of six connected rigid bodies and a seventh body summarises the human torso. The constrained multibody system is described by the configuration variable $\mathbf{q} \in \mathbb{R}^{84}$, composed by the placement of the rigid bodies' centres of mass and their right-handed director triads. As a result of the rigid body formulation, $m_{int} = 42$ internal constraints are present. The consideration of the anatomical joints cause $m_{ext} = 22$ external constraints and therefor the $k = 84$ -dimensional system is restricted by 64 holonomic constraints. Corresponding to the

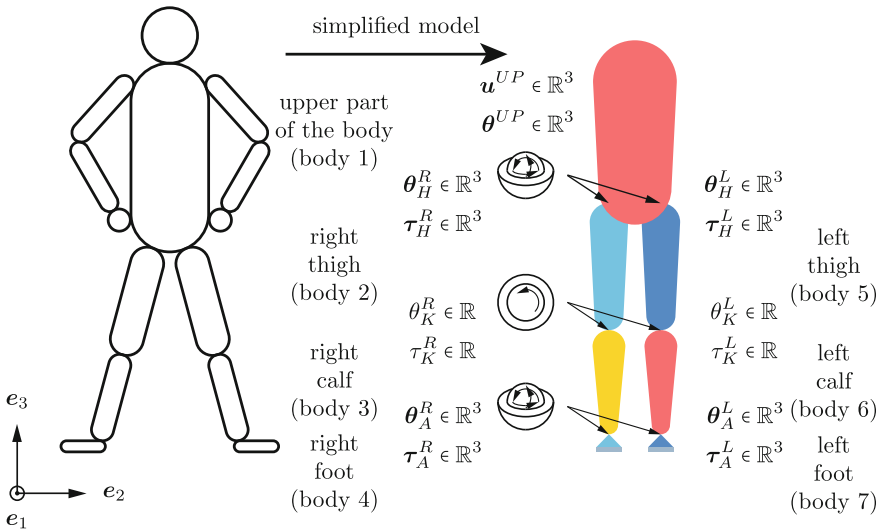


Fig. 6.1 Seven link simplified bipedal walker and its generalised coordinates

Table 6.1 Physical quantities of the human torso, thigh, calf and foot are taken from [2]

Physical quantity	Torso	Thigh	Calf	Foot
Mass (kg)	33.9946	6.5233	2.6857	0.8372
Moment of inertia				
$I_{e_1 e_1}$ (kg · m ²)	1.6194	0.1137	0.0391	0.0034
$I_{e_2 e_2}$ (kg · m ²)	1.0876	0.1158	0.0393	0.0030
$I_{e_3 e_3}$ (kg · m ²)	0.3785	0.0225	0.0029	0.0007
Length (m)	0.6644	0.4582	0.3753	0.2433
Center of mass from proximal centroid (m)	0.3470	0.1779	0.1943	0.1368

$k - m = 20$ degrees of freedom, the generalised coordinates of the seven link model read

$$\mathbf{u} = [\mathbf{u}^{UP} \ \boldsymbol{\theta}^{UP} \ \boldsymbol{\theta}_H^R \ \boldsymbol{\theta}_K^R \ \boldsymbol{\theta}_A^R \ \boldsymbol{\theta}_H^L \ \boldsymbol{\theta}_K^L \ \boldsymbol{\theta}_A^L]^T \in \mathbb{R}^{20},$$

whereby \mathbf{u}^{UP} and $\boldsymbol{\theta}^{UP}$ represent the translation respectively the rotational motion of the upper part of the human body. The walker model is only actuated in the hip, knee and ankle joints of both legs by

$$\boldsymbol{\tau} = [\boldsymbol{\tau}_H^R \ \boldsymbol{\tau}_K^R \ \boldsymbol{\tau}_A^R \ \boldsymbol{\tau}_H^L \ \boldsymbol{\tau}_K^L \ \boldsymbol{\tau}_A^L]^T \in \mathbb{R}^{14},$$

which means only internal actuation torques are applied and the system is underactuated.

To allow the different contact scenarios during the walking sequence, two contact points are necessary to model the two single contact phases (forefoot contact (FC), respectively heel contact (HC)) as illustrated in Fig. 6.2 and the complete contact support phase, at which the forefoot and the heel are in contact with the ground. Table 6.2 contains geometrical quantities of the seven-link walker model, whereby $\boldsymbol{\rho}_{HC}$ and $\boldsymbol{\rho}_{FC}$ point from the centre of mass of the foot to the heel respectively forefoot contact point and vector $\boldsymbol{\rho}_A$ points to the ankle. In contrast to the jumper model in [27], where the thigh is fixed in the middle of the torso bottom, the right leg is connected with the upper part of the body at $\boldsymbol{\rho}_H^R$ pointing from the torso's centre of mass to the hip joint—and due to the bilateral symmetry, the left thigh is fixed at $\boldsymbol{\rho}_H^L$.

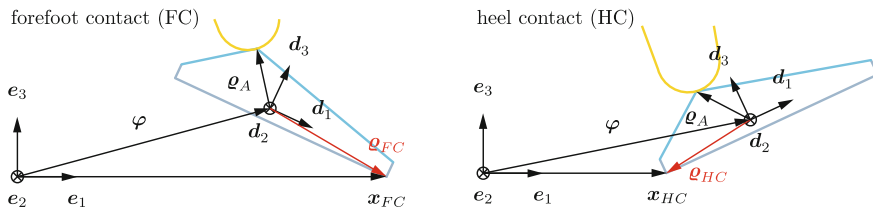


Fig. 6.2 Visualisation of the forefoot (FC) and heel contact (HC)

Table 6.2 Geometrical quantities of the feet on the basis of the results from [2] and the hips position described by the torso's centre of mass

Geometrical quantity (m)	e_1	e_2	e_3
Height of ankle	0	0	0.0635
Foot breadth	0	0.0915	0
$\boldsymbol{\rho}_A$	-0.0644	0	0.0502
$\boldsymbol{\rho}_{HC}$	-0.1030	0	-0.0133
$\boldsymbol{\rho}_{FC}$	0.1403	0	-0.0133
$\boldsymbol{\rho}_H^R$	0	-0.0936	-0.2398
$\boldsymbol{\rho}_H^L$	0	0.0936	-0.2398

According to the human anatomy, the motion capabilities of the different joints are limited by inequality path constraints which are part of the optimal control problem, see Sect. 6.7. As said before, the hips are modelled as spherical joints, whereby the principal axes of rotation coincide with the global coordinate system. The rotation around the sagittal axis e_1 and around the longitudinal axis e_3 is restricted to a range of $\pm 30^\circ$. For the hips' extension, the maximum ankle is set to 30° , while the range of the flexion motion is assumed to be 120° . The human knees are modelled as revolute joints and the previously introduced unit vectors n^2 and n^5 specify the anatomical transversal axes. In this work, the knees' flexion condition restricts the motion to 120° and the extension ankle is set to 0° such that the super extension is prevented. In contrast to the real human ankle joint, the motions of the upper and lower ankle are summarised at a single spherical joint. The rotation around the axes e_1 and e_3 is set to $\pm 30^\circ$, the dorsal extension and the plantar flexion at the sagittal plane is constrained to 30° respectively 50° . All the anatomical joint limitations are summarised as $h_{anat}(\mathbf{q}) < \mathbf{0} \in \mathbb{R}^8$.

6.6 Human Walking Gait

This section describes the sequence of human gait as a problem with multiple subsequent motion phases—see Fig. 6.3—based on a multibody system model and the achieved forward locomotion is a result of the lower extremities' movement (compare to [35]). The human gait is defined as bipedal, forward propulsion of the human body's centre of gravity and during the motion alternate sinuous movements of different body parts are observable. The different gait patterns are the result of differences in the movements of the lower extremities concerning velocity, forces, kinetic and potential energy and especially the changes of contact between foot and ground. Natural gaits are classified as meander, walk, jog, run and sprint with increasing order of speed [48]. Generally, the natural gaits are designed to propel a person in forward direction, but they can also be adopted for lateral movements.

We are interested in the walking movement along a straight line and it is assumed, that the right and left steps are identical such that periodicity constraints on the posture, the velocities and actuation can be defined. Analogous to the compass gait in [32], a mirror function in conjunction with a mirror plane is introduced to guarantee that the final state is a mirrored image of the initial state and it is displaced in walking direction, compare the left and right photo of Fig. 6.3. The mirror plane is placed in the middle of both heel contact points and it is spanned by the vertical unit vector and the walking direction. Let *mirr* denote the mirror function, then it is possible to model the periodicity constraints for the legs as mirror images of the configurations and control variables in the form

$$\mathbf{q}_0 = \text{mirr}(\mathbf{q}_N), \quad \mathbf{q}_1 = \text{mirr}(\mathbf{q}_{N+1}), \quad \boldsymbol{\tau}_0 = \text{mirr}(\boldsymbol{\tau}_N) \quad (6.4)$$

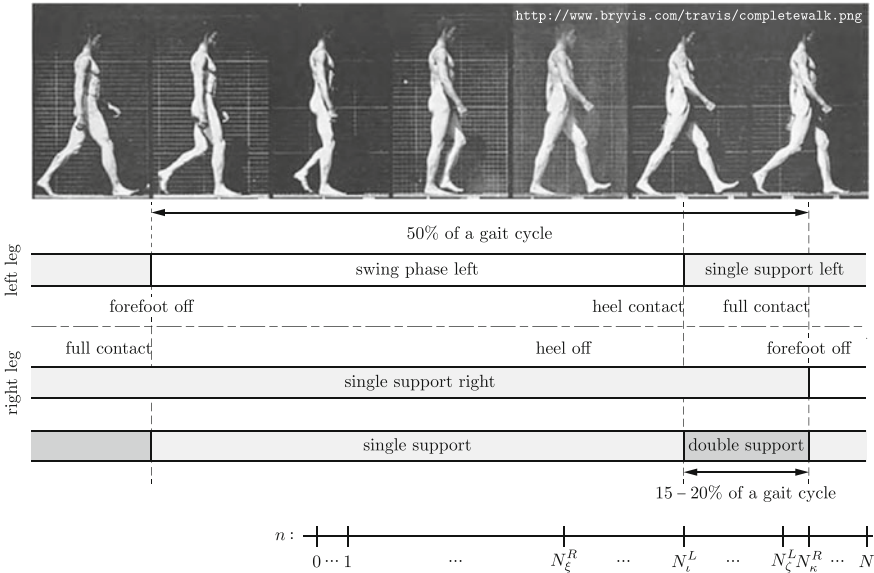


Fig. 6.3 Photographies of human walk cycle for a single step and illustration of the support phases

At this point it should be mentioned that instead of the configuration based mirror conditions of the time nodes in Eq. (6.4), the periodic boundary conditions can also be realised by mirror images of the leg’s configurations and the conjugate momenta at the boundary time nodes, as it is illustrated in Fig. 6.4. According to the sequence of forefoot and heel contact releasing and establishing in Fig. 6.3, the human gait motion has to be modelled in a correct order in the continuous optimal control problem as well as in the discrete optimisation problem. In both cases, the double support phase represents the essential challenges of the human walking phases due to the occurring closed loop conditions. Compared with running, the walking movement is characterised by the chronology of single and double support phases and thus at any given time at least one foot is in contact with the ground: no flight phase occurs as can be observed in running motions. As illustrated in Figs. 6.3, 6.4 and 6.5, the optimal control problem of the half gait cycle is discretised using N time nodes and the gait cycle starts with single support whereby the right foot is in full contact. For the optimisation we assume that the heel contact at the right foot is released at the node number N_{ξ}^R and the double support phase is initiated by left heel contact at the time node N_{ι}^L . Then, the left legs forefoot gets in contact with the ground at node N_{ζ}^L , thus the second part of double support is realised: namely, the left foot is in full contact while at the same time the right foot is in forefoot contact with the ground. The double support phase ends at node number N_{κ}^R , at which the forefoot contact of the right foot is released.

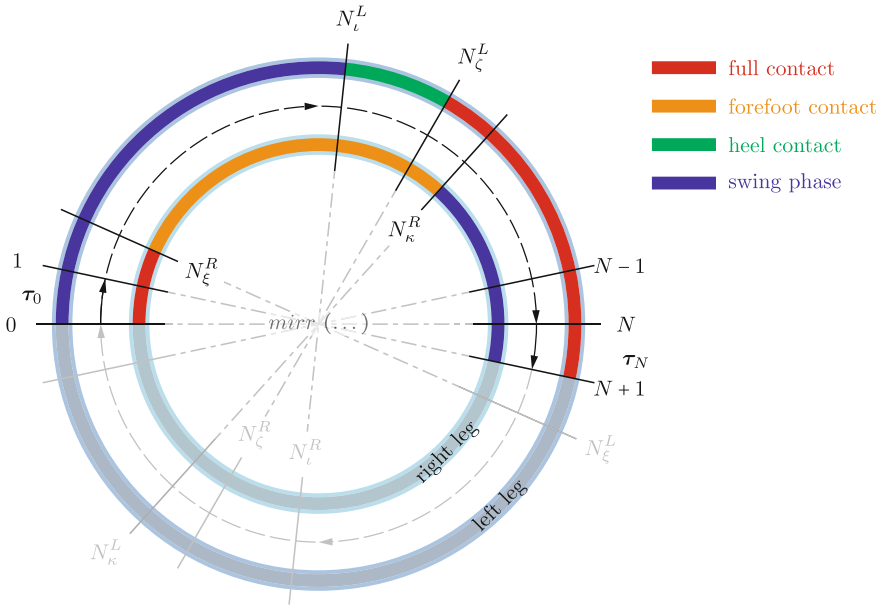


Fig. 6.4 Illustration of the gait cycle periodicity and the motion sequence of the discrete optimal control problem with periodic boundary conditions

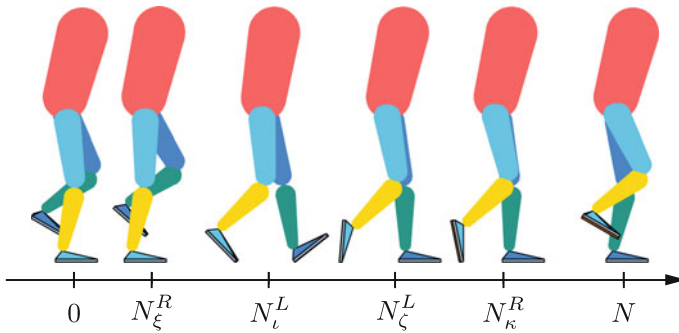


Fig. 6.5 Illustration of the characteristic configurations at the transition between two contact phases for a half gait cycle

6.7 Optimal Control of the Bipedal Walker

The constrained optimisation problem is formulated in terms of generalised coordinates \mathbf{u}_d and the actuation torques $\boldsymbol{\tau}_d$. In case of free motion (no contacts occur), the dynamical system is restricted by the internal and external constraints in order to model the rigidity of the bodies and the structural composition of the multibody system itself and the dynamic is described by a 20-dimensional system of equations. In all

five walking sections, the same relative nodal reparametrisation is used to update the configuration in dependence of the generalised coordinates. The equations of motion during the walking phases are premultiplied with the appropriate single respectively full contact discrete null space matrices to eliminate the constraint forces including the contact forces. Contacts between the feet and the ground are modelled as perfectly plastic contact and the orientation of the contact forces prevent penetrating the ground. As a result of the different contact scenarios during the walking phases, the contact Lagrange multipliers are determined in a post-processing process via

$$\lambda_{\Gamma,n} = \mathbf{R}_{\Gamma}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^-],$$

where

$$\mathbf{R}_{\Gamma}(\mathbf{q}_n) = \mathbf{G}_{\Gamma}^T(\mathbf{q}_n) \cdot (\mathbf{G}_{\Gamma}(\mathbf{q}_n) \cdot \mathbf{G}_{\Gamma}^T(\mathbf{q}_n))^{-1} \quad \text{for} \quad \Gamma = \text{FHC, HC, FC,}$$

which is formulated with the discrete Jacobian matrix $\mathbf{G}_{\Gamma}(\mathbf{q}_n)$ of the seven-link model for the active contact scenarios (full contact (FHC), forefoot contact (FC) and heel contact (HC)). Furthermore, during the double support phases, the Coulomb's static friction law has to be fulfilled at the contact points. Consequently two inequality constraints are necessary.

The determination of the respective null space matrices for the double support phase is avoided by taking a similar approach as for the transfer of contact at the compass gait in [32]. The first part of the double support phase is characterised by the forefoot contact of the right leg and by the heel contact of the left leg as illustrated in Fig. 6.4. At first, the Euler–Lagrange equations are premultiplied by the transposed discrete null space matrix ${}^R\mathbf{P}_{FC}$ to eliminate the contact forces at the right forefoot. Afterwards we premultiply the partially reduced equations of motion again by the following projection matrix ${}^L\mathbf{Q}_{HC} \in \mathbb{R}^{17 \times 17}$ with

$${}^L\mathbf{Q}_{HC} = \mathbf{I} - \left({}^L\mathbf{G}_{HC} \cdot {}^R\mathbf{P}_{FC} \right)^T \cdot \left(\left({}^L\mathbf{G}_{HC} \cdot {}^R\mathbf{P}_{FC} \right) \cdot \left({}^L\mathbf{G}_{HC} \cdot {}^R\mathbf{P}_{FC} \right)^T \right)^{-1} \cdot {}^L\mathbf{G}_{HC} \cdot {}^R\mathbf{P}_{FC}. \quad (6.5)$$

At the second part of the double support phase—forefoot contact on the right foot and full contact on the left leg—the equations of motion are premultiplied at first by ${}^L\mathbf{P}_{FHC}^T$ and then by the matrix ${}^R\mathbf{Q}_{FC} \in \mathbb{R}^{14 \times 14}$ given by

$${}^R\mathbf{Q}_{FC} = \mathbf{I} - \left({}^R\mathbf{G}_{FC} \cdot {}^L\mathbf{P}_{FHC} \right)^T \cdot \left(\left({}^R\mathbf{G}_{FC} \cdot {}^L\mathbf{P}_{FHC} \right) \cdot \left({}^R\mathbf{G}_{FC} \cdot {}^L\mathbf{P}_{FHC} \right)^T \right)^{-1} \cdot {}^R\mathbf{G}_{FC} \cdot {}^R\mathbf{P}_{FHC}. \quad (6.6)$$

Due to the unknown switching times between the different walking sequences, four scaling parameters summarised in $\boldsymbol{\sigma} = [\sigma_1 \ \sigma_2 \ \sigma_3 \ \sigma_4]^T \in \mathbb{R}^4$ are required and they are also part of the optimisation variables. It is necessary to mention that the scaling parameter σ_1 is used twice, namely at the single support phases at the beginning and at the end of the optimisation problem. These parameters enable the optimiser to shorten or extend each walking phase inside the scaling limits. Finally, the discrete

constrained optimisation problem of the half cycle gait for the chosen conditions reads

$$\min_{\mathbf{u}_d, \boldsymbol{\tau}_d, \boldsymbol{\sigma}} J(\mathbf{u}_d, \boldsymbol{\tau}_d, \boldsymbol{\sigma}) = \sum_{n=0}^N C(\mathbf{u}_n, \boldsymbol{\tau}_n, t_n, t_{n+1})$$

subject to the reduced forced discrete equations of motion of the

- single support phase with full contact in $[t_0, t_\xi]$ for the right foot ($\varepsilon = R$) and in $[t_\kappa, t_N]$ for the left foot ($\varepsilon = L$)

$$\begin{aligned} {}^\varepsilon \mathbf{P}_{FHC}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^-] &= \mathbf{0} \\ {}^\varepsilon \mathbf{g}_{FHC}(\mathbf{q}_{n+1}) &= \mathbf{0} \end{aligned}$$

- single support phase with forefoot contact in $]t_\xi, t_l[$ for the right foot

$$\begin{aligned} {}^R \mathbf{P}_{FC}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^-] &= \mathbf{0} \\ {}^R \mathbf{g}_{FC}(\mathbf{q}_{n+1}) &= \mathbf{0} \end{aligned}$$

- double support phase with forefoot contact of the right foot and heel contact of the left foot in $[t_l, t_\zeta]$

$$\begin{aligned} {}^L \mathbf{Q}_{HC} \cdot {}^R \mathbf{P}_{FC}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^-] &= \mathbf{0} \\ {}^R \mathbf{g}_{FC}(\mathbf{q}_{n+1}) &= \mathbf{0} \\ {}^L \mathbf{g}_{HC}(\mathbf{q}_{n+1}) &= \mathbf{0} \end{aligned}$$

- double support phase forefoot contact of the right foot and full contact at the left leg in $]t_\zeta, t_\kappa[$

$$\begin{aligned} {}^R \mathbf{Q}_{FC} \cdot {}^L \mathbf{P}_{FHC}^T(\mathbf{q}_n) \cdot [D_2 \mathcal{L}_d(\mathbf{q}_{n-1}, \mathbf{q}_n) + D_1 \mathcal{L}_d(\mathbf{q}_n, \mathbf{q}_{n+1}) + \mathbf{f}_{n-1}^+ + \mathbf{f}_n^-] &= \mathbf{0} \\ {}^R \mathbf{g}_{FC}(\mathbf{q}_{n+1}) &= \mathbf{0} \\ {}^L \mathbf{g}_{FHC}(\mathbf{q}_{n+1}) &= \mathbf{0} \end{aligned}$$

- periodic boundary conditions (6.4) are summarised by

$$\mathbf{b}(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_N, \mathbf{q}_{N+1}, \boldsymbol{\tau}_0, \boldsymbol{\tau}_N) = \mathbf{0}$$

- path constraints for $[t_0, t_N]$

$$\mathbf{h}_{eq}(\mathbf{q}_n) = \mathbf{0} \quad \mathbf{h}_{ineq}(\mathbf{q}_n) < \mathbf{0}$$

- anatomical angle conditions of the different joints for $[t_0, t_N]$

$$\mathbf{h}_{anat}(\mathbf{q}_n) < \mathbf{0}$$

- path constraints single support

- full contact for $n = 1, \dots, N_\xi^R$ and $n = N_\kappa^R + 1, \dots, N - 1$
right leg $n = 1, \dots, N_\xi^R - 1$

$$\sqrt{(\lambda_{FHC,n}^1)^2 + (\lambda_{FHC,n}^2)^2} - \mu_0 |\lambda_{FHC,n}^3 + \lambda_{FHC,n}^6| < 0 \wedge \lambda_{FHC,n}^3 < 0 \wedge \lambda_{FHC,n}^6 < 0$$

right leg $n = N_\xi^R$

$$\lambda_{FHC,n}^4 = \lambda_{FHC,n}^5 = \lambda_{FHC,n}^6 = 0 \wedge \sqrt{(\lambda_{FHC,n}^1)^2 + (\lambda_{FHC,n}^2)^2} - \mu_0 |\lambda_{FHC,n}^3| < 0 \wedge \lambda_{FHC,n}^3 < 0$$

left leg $n = N_\kappa^R + 1, \dots, N - 1$

$$\sqrt{(\lambda_{FHC,n}^1)^2 + (\lambda_{FHC,n}^2)^2} - \mu_0 |\lambda_{FHC,n}^3 + \lambda_{FHC,n}^6| < 0 \wedge \lambda_{FHC,n}^3 < 0 \wedge \lambda_{FHC,n}^6 < 0$$

- forefoot contact for $n = N_\xi^R + 1, \dots, N_\iota^L - 1$
right leg

$$\sqrt{(\lambda_{FC,n}^1)^2 + (\lambda_{FC,n}^2)^2} - \mu_0 |\lambda_{FC,n}^3| < 0 \wedge \lambda_{FC,n}^3 < 0$$

- path constraints double support

- single/single contact for $n = N_\iota^L, \dots, N_\zeta^L - 1$
left foot

$$\sqrt{(\lambda_{HC,n}^1)^2 + (\lambda_{HC,n}^2)^2} - \mu_0 |\lambda_{HC,n}^3| < 0 \wedge \lambda_{HC,n}^3 < 0$$

right foot

$$\sqrt{(\lambda_{FC,n}^1)^2 + (\lambda_{FC,n}^2)^2} - \mu_0 |\lambda_{FC,n}^3| < 0 \wedge \lambda_{FC,n}^3 < 0$$

- single/full contact for $n = N_\zeta^L, \dots, N_\kappa^R$
left leg $n = N_\zeta^L, \dots, N_\kappa^R$

$$\sqrt{(\lambda_{FHC,n}^1)^2 + (\lambda_{FHC,n}^2)^2} - \mu_0 |\lambda_{FHC,n}^3 + \lambda_{FHC,n}^6| < 0 \wedge \lambda_{FHC,n}^3 < 0 \wedge \lambda_{FHC,n}^6 < 0$$

right leg for $n = N_\zeta^L, \dots, N_\kappa^R - 1$

$$\sqrt{(\lambda_{FC,n}^1)^2 + (\lambda_{FC,n}^2)^2} - \mu_0 |\lambda_{FC,n}^3| < 0 \wedge \lambda_{FC,n}^3 < 0$$

right leg $n = N_\kappa^R$ $\lambda_{FC,n}^1 = \lambda_{FC,n}^2 = \lambda_{FC,n}^3 = 0$

- scaling parameters

$$\sigma_{LB} \leq \sigma \leq \sigma_{UB}.$$

The duration of each phase is scaled by its appropriate scaling parameter, whereby due to the periodicity the scaling parameter σ_1 is used twice, namely at the single support phases at the beginning and at the end of the optimisation problem. Analogous to the optimisation of the jumping movement in [27], the scaling parameter bounds σ_{LB}, σ_{UB} yield a lower and upper limit of the manoeuvre time for the half step. The characteristic configurations representing the switching points between the different contact phase are depicted in Fig. 6.5 during a half gait cycle.

6.7.1 Numerical Results for the Half Gait Cycle

The optimal control problem of the half gait cycle is discretised using $N = 16$ time nodes and the time nodes separating the different gait phases are chosen as $N_\xi^R = 3$, $N_t^L = 9$, $N_\zeta^L = 11$ and $N_\kappa^R = 12$. The choice of the optimal periodic boundary configurations is left to the optimiser. Besides the anatomical constraints for both legs, the adherence of Coulomb's static friction law generates further inequality constraints (with a static friction coefficient $\mu_0 = 0.7$). The allowed stiction range is symbolised by the cross hatched in the following illustrations Figs. 6.9, 6.14, 6.19 and 6.24 in which the contact Lagrange multipliers are plotted over time.

The basic time step of $\Delta t = 0.0250$ s is used to discretise the half gait cycle, where each scaling parameter for the different manoeuvre phases is bounded between $0.5 \leq \sigma_1, \sigma_2, \sigma_3, \sigma_4, \leq 1.5$. The optimisation problems are solved in MATLAB using the `sqp` algorithm of the `fmincon` function and a numerical accuracy is set to 10^{-8} . Finally, the initial guess is a 568-dimensional vector, which consist of 340 discrete generalised coordinates, 224 generalised internal actuation torques and the four scaling parameters. The optimisation problem is solved for the objectives of minimal specific control effort, minimal specific torque change, minimal specific kinetic energy and minimal specific jerk. As mentioned in context of the bipedal compass gait in [32], the objective function per step length $sl \in \mathbb{R}$ in \mathbf{e}_2 -walking direction is called specific objective function. At first, we choose the widely used criterion of minimal control effort [8, 29] per step length, which can be formulated as

$$J_d(\mathbf{u}_d, \boldsymbol{\tau}_d, \boldsymbol{\sigma}) = \frac{1}{2sl(\mathbf{u}_d, \boldsymbol{\tau}_d)} \sum_{n=0}^{N-1} (t_{n+1} - t_n) |\boldsymbol{\tau}_n|^2, \quad (6.7)$$

to achieve a minimally exhausting motion. Another objective function accounts the minimisation of change in torques per step length during the simulation (see [36, 49, 52]), and reads

$$J_d(\mathbf{u}_d, \boldsymbol{\tau}_d, \boldsymbol{\sigma}) = \frac{1}{2sl(\mathbf{u}_d, \boldsymbol{\tau}_d)} \sum_{n=0}^{N-1} (t_{n+1} - t_n) \left| \frac{\boldsymbol{\tau}_{n+1} - \boldsymbol{\tau}_n}{(t_{n+1} - t_n)} \right|^2. \quad (6.8)$$

The physiological motivation for this criterion is based on two different aspects: the first one states that the arbitrary fast change in torques is physiologically not possible and secondly one reduces the control interventions of the central nervous system. Hence, the resulting smooth evolution of actuation torques is probably more realistic than that resulting from the criterion of minimal control effort. In addition to the criteria of minimal control effort and minimal torque change in (6.7) and (6.8) respectively, a third aim is to optimise bipedal walking with minimal kinetic energy (similar to [9, 18–20, 46, 50]). The objective function of kinetic energy is formulated using the constant symmetric mass matrix \mathbf{M} and the projected conjugate momentum $\mathcal{Q}\mathbf{p}_n^+$ (being consistent with the holonomic constraints on momentum level see [31]), thus the specific objective function is given as

$$J_d(\mathbf{u}_d, \boldsymbol{\tau}_d, \boldsymbol{\sigma}) = \frac{1}{2sl(\mathbf{u}_d, \boldsymbol{\tau}_d)} \sum_{n=1}^N (t_{n+1} - t_n) \left((\mathcal{Q}\mathbf{p}_n^+)^T \cdot \mathbf{M}^{-1} \cdot \mathcal{Q}\mathbf{p}_n^+ \right),$$

with the projection matrices $\mathcal{Q}(\mathbf{q}_n)$ corresponding to the different gait phases similar to the projection matrices in (6.5) and (6.6). A physiological motivation for this objective function is that a motion consuming as little energy as possible can be practised over a longer period of time. Additionally, the optimisation problem is solved for minimal jerk per step length, this objective function is given as

$$J_d(\mathbf{u}_d, \boldsymbol{\tau}, \boldsymbol{\sigma}) = \frac{1}{2sl(\mathbf{u}_d, \boldsymbol{\tau}_d)} \sum_{n=2}^N (t_n - t_{n-1}) \left(\mathbf{s}_n''' \right)^T \cdot \mathbf{s}_n''',$$

with \mathbf{s}_n''' being the third derivative of the trajectory. The discrete jerk is calculated with finite differences and the criterion is formulated in relative generalised coordinates of all joints

$$\mathbf{s}_n''' = \frac{\mathbf{s}_{n+\frac{1}{2}}'' - \mathbf{s}_{n-\frac{1}{2}}''}{\frac{1}{2}(t_{n+1} - t_{n-1})} = \frac{1}{\frac{1}{2}(t_{n+1} - t_{n-1})} \left(\frac{\frac{\mathbf{u}_{n+1}}{t_{n+1}-t_n} - \frac{\mathbf{u}_n}{t_n-t_{n-1}}}{t_{n+1} - t_n} - \frac{\frac{\mathbf{u}_n}{t_n-t_{n-1}} - \frac{\mathbf{u}_{n-1}}{t_{n-1}-t_{n-2}}}{t_n - t_{n-1}} \right).$$

Generally, the criterion of minimal jerk is chosen to yield uniformly accelerated motions.

Note that during the different walking phases we distinguish between full, forefoot and heel contact sequences and the different occurring contacts are symbolised by the corresponding contact Lagrange multipliers ($\boldsymbol{\lambda}_{FHC} \in \mathbb{R}^6$, $\boldsymbol{\lambda}_{FC} \in \mathbb{R}^3$ and $\boldsymbol{\lambda}_{HC} \in \mathbb{R}^3$). Consequently, twelve different contact Lagrange multipliers are the result of the simulated half gait cycle. To avoid a confusing evaluation of the contact Lagrange multipliers, we decide to introduce the forefoot respectively heel related Lagrange multipliers $^{fore}\boldsymbol{\lambda}_C \in \mathbb{R}^3$ and $^{heel}\boldsymbol{\lambda}_C \in \mathbb{R}^3$. According to each walking phase—highlighted via different shades of grey—and with the knowledge of the investigated course of motions, it is easy to analyse which kind of contact is present and to read off the computed contact forces.

6.7.1.1 Minimal Specific Control Effort

In Fig. 6.6, some snapshots are depicted to represent the essential configurations of the bipedal gait for the criterion of minimal control effort per step-length. A step length of 0.5877 m is achieved and a control effort of $8.8282 \times 10^3 \text{ (Nm)}^2 \text{ s}$ is calculated for the half gait cycle inclusive the two time nodes of the subsequent half gait cycle to formulate the periodic boundary conditions (see Fig. 6.4). Thereby it is necessary to mention that the control effort of each leg is weighted by the vector

$$\mathbf{w}^I = [\sqrt{10} \ 1 \ \sqrt{10} \ 1 \ \sqrt{10} \ 1 \ \sqrt{10}]^T \in \mathbb{R}^7 \quad \text{for } I = L, R.$$

These weighting coefficients are useful to scale the generalised actuation torques with respect to the walking direction in e_2 -direction. According to the weighting coefficients, the actuation in e_1, e_3 -directions are premultiplied by a factor larger than one and consequently the control effort around these actuation directions increase. This implies for the optimisation problem that actuations in e_2 -direction should be preferred and actuations around the e_1, e_3 -directions are penalised.

The side-view in Fig. 6.6 illustrates that the trajectory of the swing foot is divided into a take-off, a contact-less swinging and a landing phase. The take-off phase starts as soon as the heel contact is realised and is continued until the forefoot contact is released. At the moment the forefoot contact is released, the swinging phase is initialised and ends when the heel contact is established again. The landing phase ranges until the foot is in full contact. The optimised solution shows that the swinging foot is raised up until the maximal level is achieved, afterwards the swinging foot's remaining momentum as well as gravity is used to propel the foot in forward direction until the heel hits the ground, see Fig. 6.7.

The swing phases, the single and the double support phases of each foot are highlighted using different shades of grey in order to evaluate the actuation sequence and the Lagrange multipliers over time in an efficient manner. The grey dashed lines represent switching points between two different contact phases of each foot while

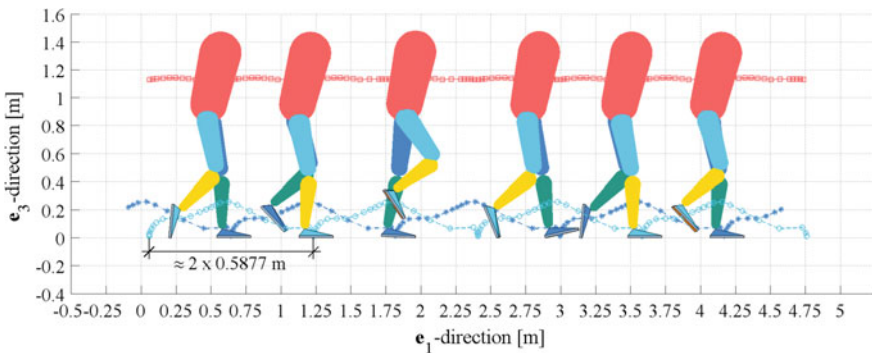


Fig. 6.6 Minimal specific control effort: snapshots of the optimised bipedal motion

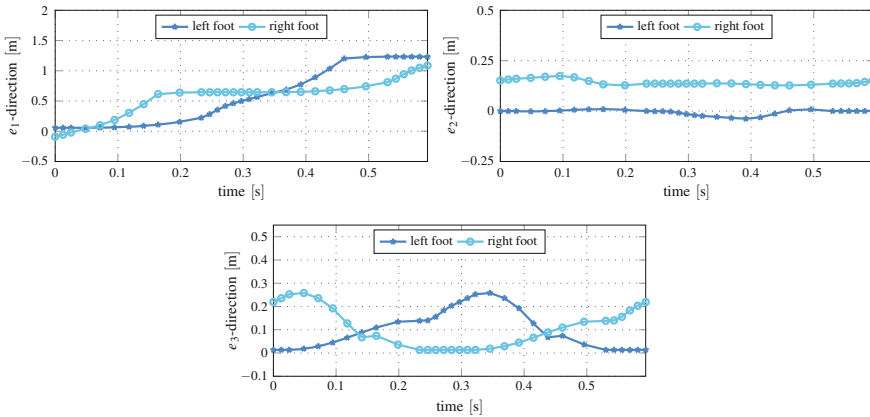


Fig. 6.7 Minimal specific control effort: evolution of the feet centre of mass' trajectory coordinates during a gait cycle

the black dashed line describes the end of the half gait cycle. This procedure is also applied for the other specific objective criteria.

The actuation torques are illustrated over time in Fig. 6.8 and it is particularly noticeable that the leg is mainly actuated by torques around the desired e_2 -direction. This behaviour is expected (not only due to the weighting components), because the step-length is measured in e_1 -direction and consequently a higher actuation around the e_2 -axis yields to a larger step-length. Furthermore, the actuation of the hip joint at the stance foot and the knee joint of the swing leg guarantee the observed walking propulsion and the uplifting motion, see the optimised torque evolution in e_2 -direction.

The scaling parameters are computed as $\sigma = [0.5058 \ 0.9261 \ 1.3803 \ 0.5273]^T$ and consequently a period of time for a half step of about 0.2970 s results, whereby a double support over a period of 0.0822 s results which correlates to 13.83 % of the gait cycle. Consequently, the optimiser shortens the double support phase compared to the observed period of real human walking, see also [16]. As indicated by the third scaling parameter, the period of double support with forefoot contact of the right leg and heel contact of the left leg is enlarged, while the last part of the double support is shortened by a scaling parameter of 0.5273.

The corresponding contact Lagrange multipliers for both legs are depicted in Fig. 6.9. Particular striking is that this optimised motion has two phases in which the contact force plays a dominant role, namely at first as soon as the left heel touches the ground and secondly during the single support phase with full contact. This behaviour can probably be explained by the fact that during the single support phase, the swinging leg is moved in forward direction and simultaneously the foot without contact is lifted up. Afterwards, the foot's potential energy is transformed in a forward motion, which ends with the heel-ground contact. As visualised in Fig. 6.9,

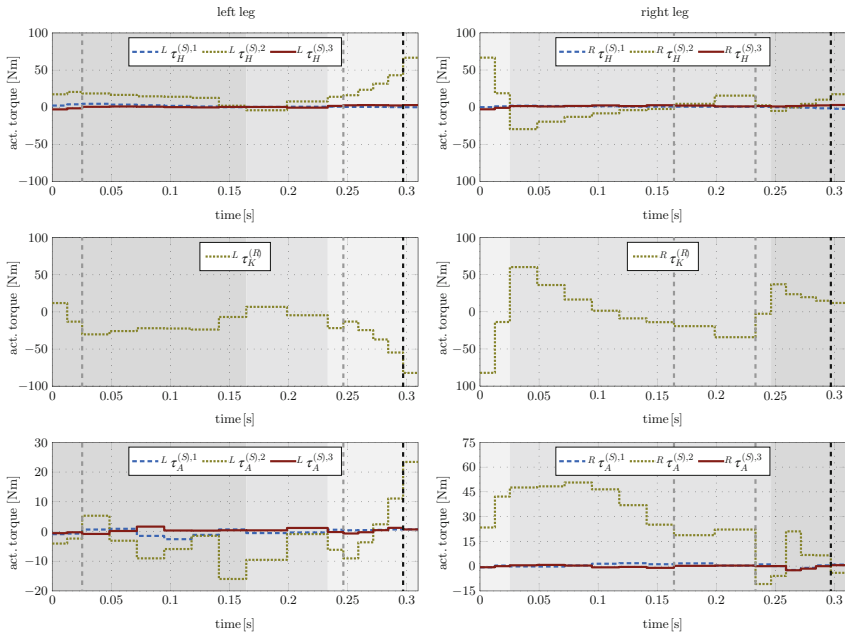


Fig. 6.8 Minimal specific control effort: actuation torques of the *left* and *right leg* during the half gait cycle

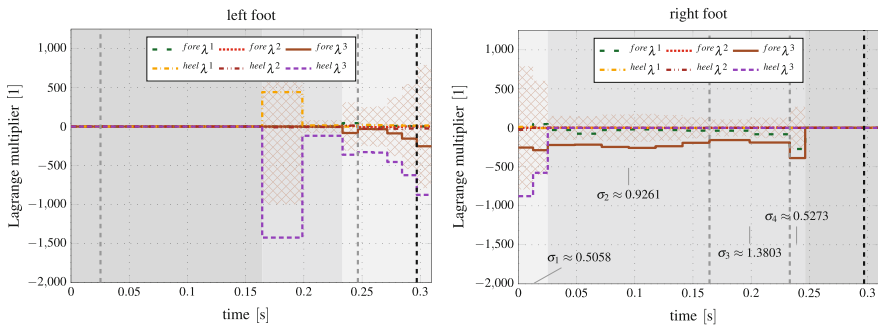


Fig. 6.9 Minimal specific control effort: evolution of the contact Lagrange multipliers of the *left* and *right leg* during the half gait cycle

the cross hatched areas represent Coulomb's static friction law ($\mu_0 = 0.7$), which is part of the inequality constraints of the optimal control problem.

The evolution of the angular momentum \mathbf{L} and of the different energy terms are depicted in Fig. 6.10. The left-hand side shows that the walker moves primarily in the $(\mathbf{e}_1, \mathbf{e}_3)$ -plane, because L_1 and L_3 are almost zero. This behaviour is accentuated by the evolution of the feet trajectory in \mathbf{e}_2 -direction of Fig. 6.7 which is small compared to the motion in \mathbf{e}_1 and \mathbf{e}_3 -direction. On the right-hand side of Fig. 6.10, the energy

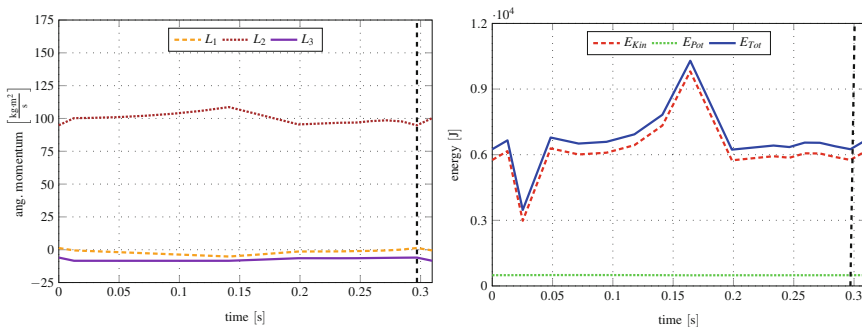


Fig. 6.10 Minimal specific control effort: evolution of the angular momentum (*left*) and the kinetic, potential and total energy (*right*) during the half gait cycle

evolution is illustrated over time, whereby kinetic energy is considerably greater than the potential energy, which is in comparison to the kinetic energy nearly constant. This is due to the small vertical movement of the walker during the bipedal gait.

6.7.1.2 Minimal Specific Torque Change

The criterion of minimising the change of torques per step length yields smoother evolutions of the joint actuations, see Fig. 6.13, and a step-length of 0.6450 m results. The torque change for the optimised gait is computed to $1.8875 \times 10^5 \text{ (Nm)}^2 \text{ s}^{-1}$. The optimised motion sequence has certain resemblances to the results of minimal control effort per step length, especially the trajectories during the swing phases before the heel contact is established. Some differences of the trajectories are observable after the take-off phase of the swinging leg, because the leg is more bent and consequently the foot is raised higher than when using specific control effort. Subsequently, the potential energy is transformed into a forward motion with a step-length of about 9 % larger then when using the previous objective criterion. This described behaviour is depicted in Fig. 6.11 as well as in the lower row of Fig. 6.12 for the trajectory in e_3 -direction.

Analogue to the results of monopedal jumping in [27], higher control forces result in comparison to the specific minimal control effort, especially at the ankle joint of the leg which is in contact with the ground. The complete manoeuvre takes place in 0.5001 s and time is scaled by the vector $\sigma = [1.0726 \ 1.5000 \ 1.4457 \ 0.6024]^T$ which indicates that the optimiser extends all walking phases with the exception of the double support with one foot in full contact. For this cost function, the double support phase corresponds to 18.46 % of the gait cycle and lies inside the range of double support for a realistic human motion, see [16].

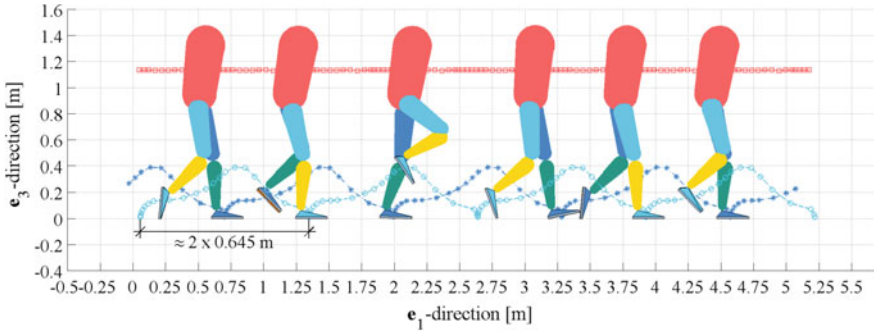


Fig. 6.11 Minimal specific torque change: snapshots of the optimised bipedal motion

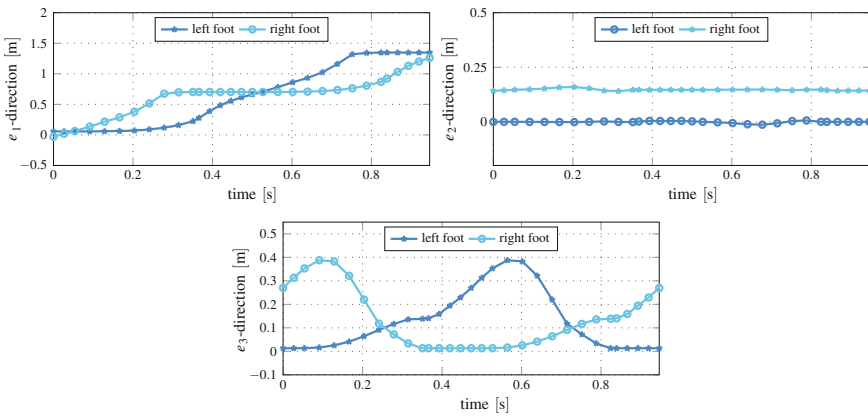


Fig. 6.12 Minimal specific torque change: evolution of the feet trajectory coordinates during a gait cycle

The evolution of the different joint torques are given in Fig. 6.13, herein also the above noticed actuation characteristics are observable. For this cost function, we chose the following scaling vector

$$w^I = [1 \ 1 \ 1 \ 1 \ \sqrt{10} \ 1 \ \sqrt{10}]^T \quad \text{for } I = L, R,$$

to increase the ankle’s influence around the e_1 and e_3 -direction on the whole motion. This is done to reduce slight motions of the feet in the (e_1, e_2) -plane. Another interesting fact is illustrated in Fig. 6.12, where the trajectory of the ankles’ e_2 -coordinate is depicted over time: namely, in contrast to the motion for the criterion of minimal control effort per step-length, resulting in relatively small ankle motions. The evolution of the e_3 -coordinate over the walking time shows the higher lifting motion. Subsequently, the achieved potential energy is transformed to a swinging motion in forward direction.

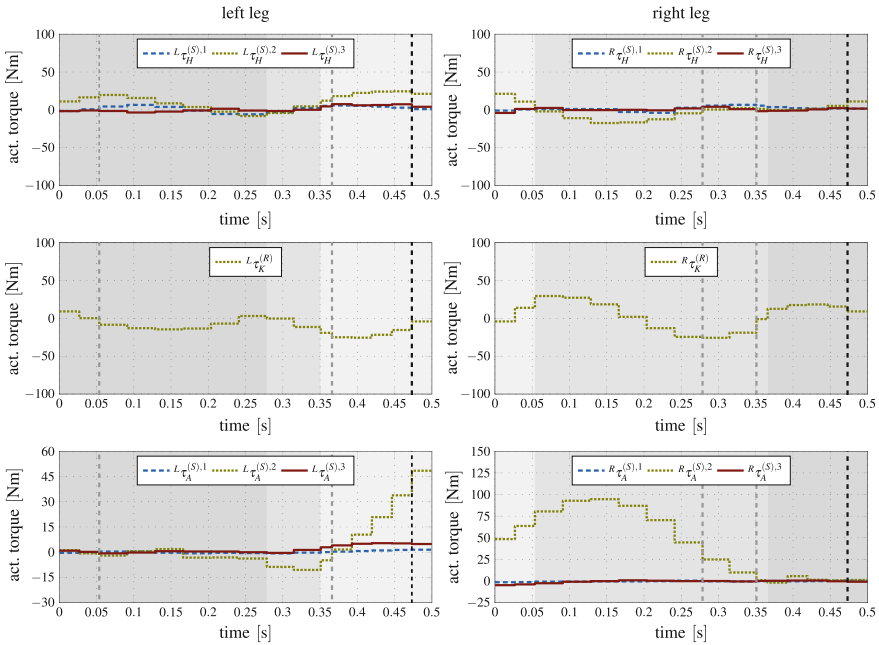


Fig. 6.13 Minimal specific torque change: actuation torques of the *left* and *right* leg during the half gait cycle

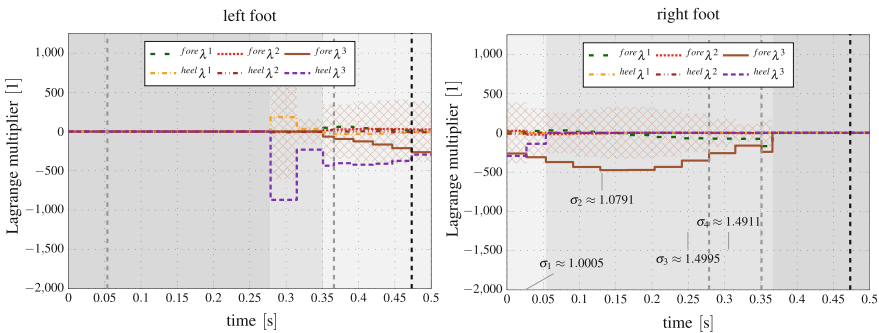


Fig. 6.14 Minimal specific torque change: evolution of the contact forces of the *left* and *right* leg during the half gait cycle

The evolution of the actuation seems more realistic if one considers the fact that an arbitrary change in torques and muscles is physiologically not possible. As mentioned before, during the walking motion, the knee is primarily actuated and the actuations in e_1 - and e_2 -direction is comparatively small. This is an interesting behaviour and it justifies the previously chosen scaling vector. The associated Lagrange multipliers for both feet are plotted in Fig. 6.14 and it is particularly noticeable that the actuation

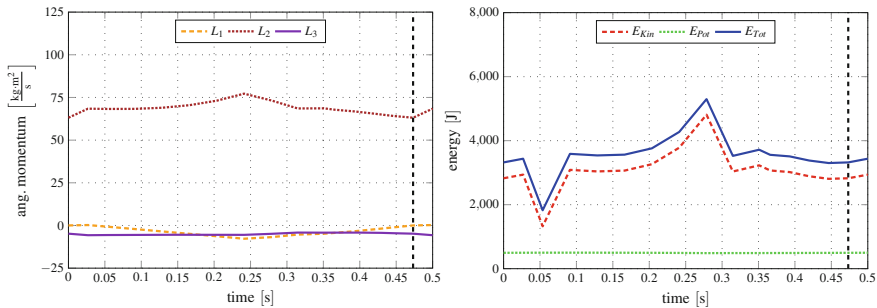


Fig. 6.15 Minimal specific torque change: evolution of the angular momentum (*left*) and the kinetic, potential and total energy (*right*) during the half gait cycle

sequences during the forefoot contact phase of the stance-foot cause a higher contact force than for the result of minimal control effort per step length. The fulfilment of the Coulomb's static friction law has a restrictive influence on the local solution.

On the left-hand side of Fig. 6.15, the evolution of angular momentum is plotted and according to the walking sequence, the different single support phases and the double support are visible. The angular momentum components L_1 and L_3 slightly oscillate and are significantly smaller than the component perpendicular to the (e_1, e_3) -walking plane. The right-hand side of Fig. 6.15 depicts kinetic, potential and total energy. We notice a qualitatively similar evolution as seen when using minimal control effort, but the kinetic energy is nearly 50 % lower than before due to the longer walking time interval.

6.7.1.3 Minimal Specific Kinetic Energy

The optimised motion for the criterion of minimal specific kinetic energy is completely different from the results for the two previous cost functions on basis of the actuation torques. The most pronounced difference concerns the trajectory of the swinging leg, which is lifted up as little as possible—the feet's center of mass is always below a height of 0.2 m (see Fig. 6.16). The kinetic energy sum for a step-length of about 0.6338 m is computed to 41.2256 J.s. That means that the optimiser extends all walking phases according to $\sigma = [1.5000 \ 1.5000 \ 1.4889 \ 1.4724]^T$, which is in all four cases exactly or almost the maximum scaling parameter. Consequently, the walking motion takes more than 0.5988 s. Another interpretation of this cost function is that we are interested in the minimal velocity per step-length, because the kinetic energy is directly proportional to the walking speed. With regard to this interpretations it is clear that a longer walking time results and the legs move primarily in e_1 -direction to achieve a larger step length.

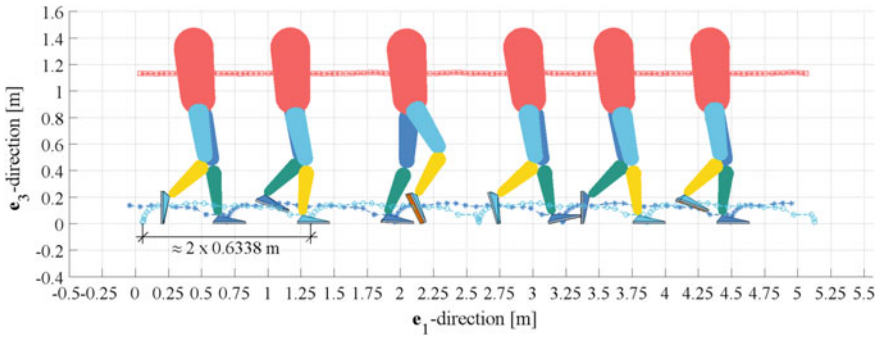


Fig. 6.16 Minimal specific kinetic energy: snapshots of the optimised bipedal motion

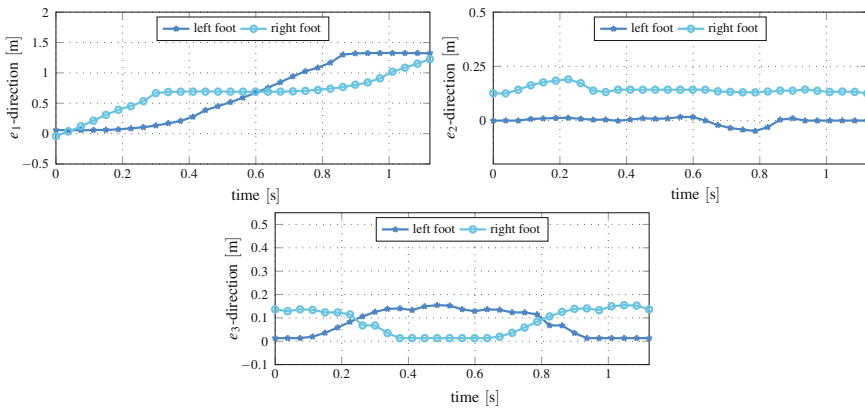


Fig. 6.17 Minimal specific kinetic energy: evolution of the feet trajectory coordinates during a gait cycle

The feet’s trajectory over time is illustrated in Fig. 6.17 and comparatively large rotational motions at the ankle joints can be observed which lead to the results illustrated in Fig. 6.17 by the trajectory in e_2 -direction. This can be explained by the fact that due to the small mass of the feet the resulting kinetic energy is small. Analogue to the motion for minimal control effort, the feet move outwards during the swing phase and also a slight horizontal motion can be noticed (Figs. 6.18 and 6.19).

As a result of the different walking behaviour, the standing leg is primarily actuated in e_2 -direction at the ankle joint during the forefoot contact. The applied actuation torques at the hip and knees have a subordinate role. Due to the lower energy during the walking motion, the contact Lagrange multipliers are smaller than observed before, whereby the Coulomb’s static friction law is also satisfied. Figure 6.20 depicts the evolution of the angular momentum and the energy during the optimised half gait cycle. The different walking phases are substantially less marked than for the criteria

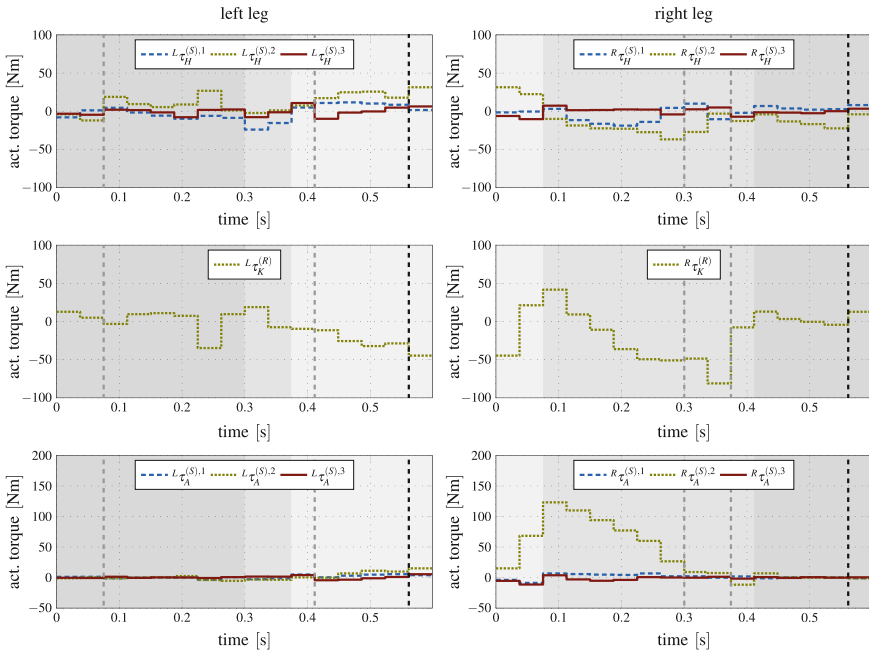


Fig. 6.18 Minimal specific kinetic energy: actuation torques of the *left* and *right* leg during the half gait cycle

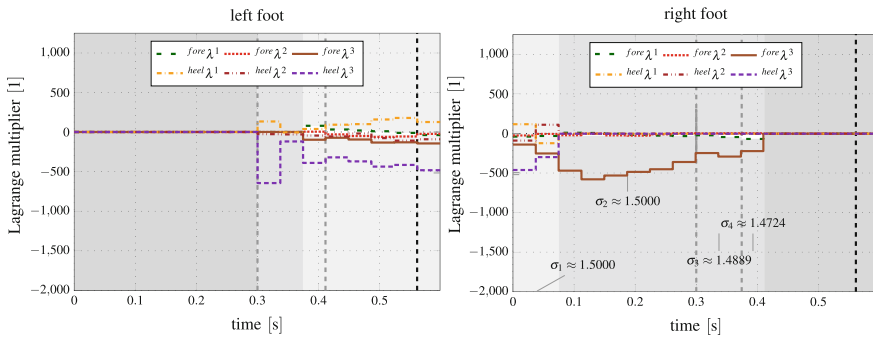


Fig. 6.19 Minimal specific kinetic energy: evolution of the contact forces of the *left* and *right* leg during the half gait cycle

before and the right-hand plot side reflects that the chosen cost function is minimising the kinetic energy. The potential energy is even smaller than before due to the minimal lifting up motion of the swinging leg. The resulting kinetic energy is substantially lower than for the other criteria.

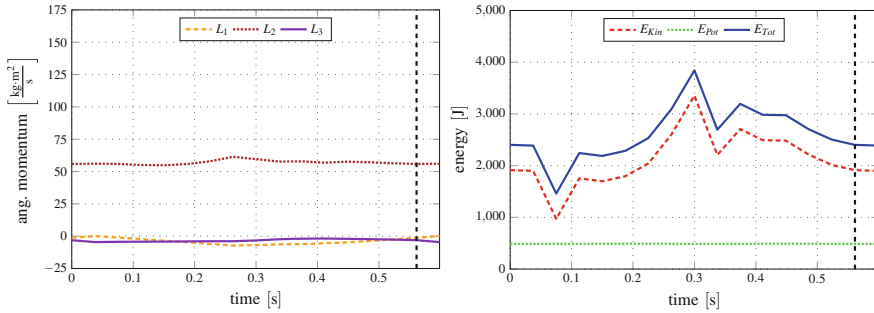


Fig. 6.20 Minimal specific kinetic energy: evolution of the angular momentum (*left*) and the kinetic, potential and total energy (*right*) during the half gait cycle

6.7.1.4 Minimal Specific Jerk

The last criterion which is discussed in this context minimises the jerk of the walker motion per step length. A step-length about 0.6247 m results and the function value is given by $1.6851 \times 10^8 \frac{\text{m}^2}{\text{s}^5}$. Comparing the trajectories of Figs. 6.11 and 6.21, the resulting foot trajectories are similar, whereby in case of a minimal jerk, the whole motion looks smoother. In contrast to the solution for minimal specific torque change, the optimiser extends all four phases. A walking time of 0.5326s arises and the double support phase corresponds 18% of one gait cycle. The optimiser extends the movements by using the scaling vector $\sigma = [1.4668 \ 1.0903 \ 1.4987 \ 1.4957]^T$.

Figure 6.22 confirms the comments respective to the similarities of both cost functions, because the evolution of the e_1 -coordinates as well as the e_2 -coordinates are very similar to those in Fig. 6.12. Also for this criterion, the feet’s trajectory is pointing outward during the swing phases. Only the vertical motion looks like a compromise of minimal specific control effort and minimal specific torque change (Fig. 6.23).

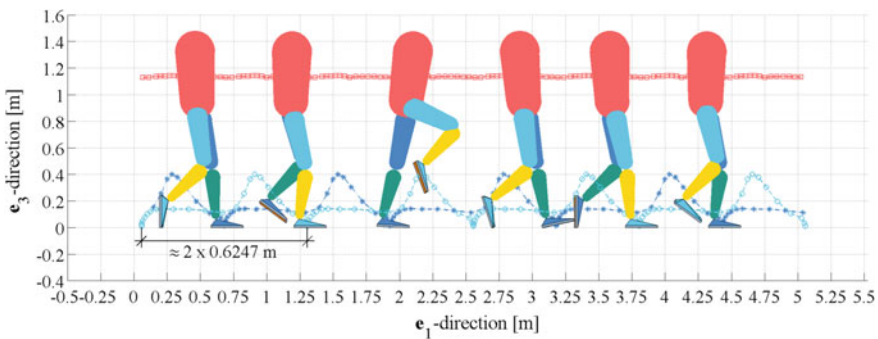


Fig. 6.21 Minimal specific jerk: snapshots of the optimised bipedal motion

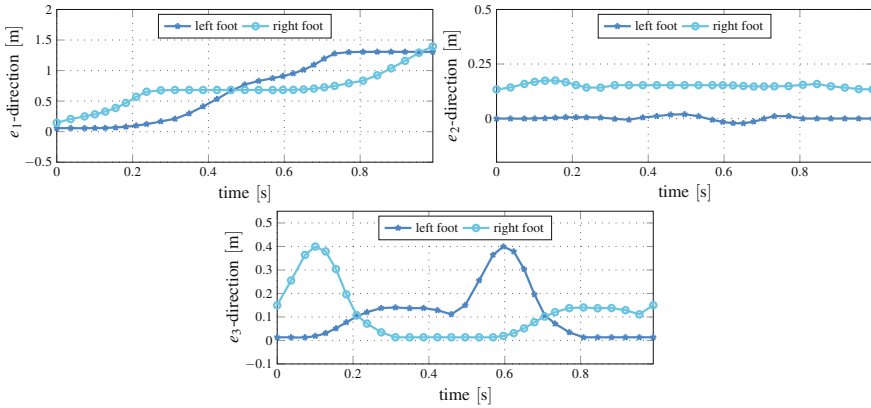


Fig. 6.22 Minimal specific jerk: evolution of the feet trajectory coordinates during a gait cycle

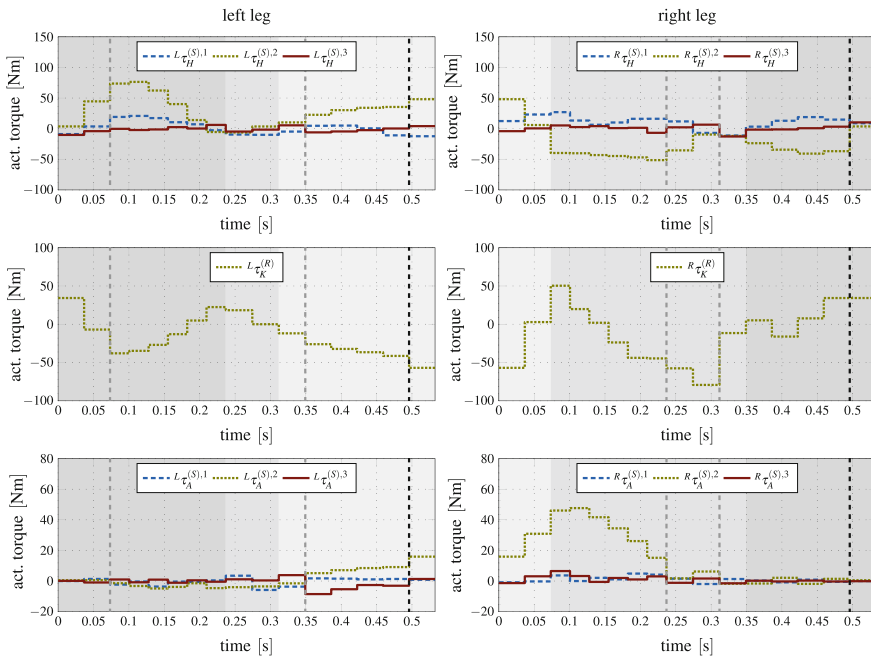


Fig. 6.23 Minimal specific jerk: actuation torques of the *left* and *right* leg during the half gait cycle

A completely different picture results if we focus on the actuation torques. The hip joints as well as the knee joints are actuated in the range ± 100 Nm, whereby especially the knee actuation reminds at the torque sequences of the criteria minimal specific torque change and minimal specific kinetic energy. Another common property between minimal energy and minimal jerk concerns the comparatively large

actuation of the stance foot’s angle joint in e_2 -direction. In the post-processing step, the resulting contact Lagrange multipliers are evaluated and depicted for both feet in Fig. 6.24. When reviewing the evolution of the contact Lagrange multipliers, it becomes evident that the computed contact Lagrange multipliers can be observed to lie between the contact force for minimal torque and minimal kinetic energy. A larger heel shock is similar to the minimal torque change and during the forefoot contact of the single support phase, the same influence of the ankle actuation becomes apparent as observed for the minimal kinetic energy solution.

Figure 6.25 visualises on the left-hand side the evolution of angular momentum. On the right-hand side the different energy terms are plotted over time. The angular momentum in e_2 -direction is very similar to the results in Fig. 6.15 and the influence of each walking phase is easily recognised. On the right-hand side, the kinetic, the potential and the total energy are depicted, whereby a qualitatively similar evolution of the different energies can also be observed as in the case of the other cost functions. Consequently, the qualitatively energy evolution is characteristic for the human walk, but the amount of energy depends on the chosen cost function.

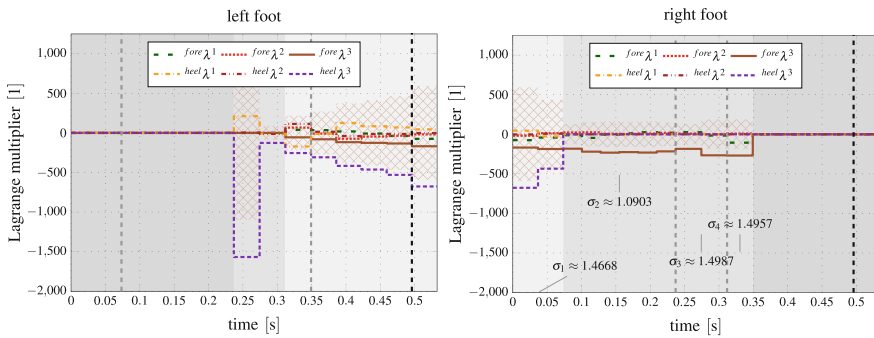


Fig. 6.24 Minimal specific jerk: evolution of the contact forces of the *left* and *right leg* during the half gait cycle

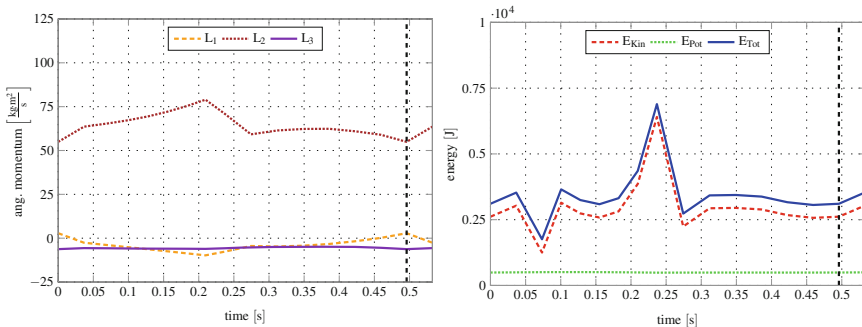


Fig. 6.25 Minimal specific jerk: evolution of the angular momentum (*left*) and the kinetic, potential and total energy (*right*) during the half gait cycle

6.8 Conclusions and Outlook on Future Work

In this work, a multibody system, which consists of seven rigid bodies, is used to investigate the bipedal upright gait in a optimal control simulation. The investigated multibody system is deliberately kept simple to purely analysing the changing dynamical behaviour during the bipedal walking with feet. The contact between the feet and the ground during the support phases is modelled by a perfectly plastic contact formulation. The optimal control problem of the human gait is solved by a direct transcription method to transform it into an optimisation problem, whereby the goal is to minimise the cost function per step length. The dynamics of the multibody system is discretised using a variational integrator with the described structure preserving properties. The influence of the different specific objective functions is investigated. Especially the optimised solution of specific kinetic energy shows great correspondence to human walking and corroborates the thesis that the human's upright gait is an anthropological optimised solution of walking with minimal kinetic energy—namely with great perseverance. These similarities to real motions accentuate the DMOCC's potential on future work. For this purpose, several options are possible: comparison with measured walking motions using the motion capturing method and the identification of possible weighting components for the specific cost functions. Further purposes may be to investigate the influence of swinging arms and to enable motion capabilities inside the model's backbone. Furthermore it might be interesting to optimise the bipedal gait along an inclined plane or to investigate climbing stairs.

References

1. Ackermann M, Schiehlen W (2006) Dynamic analysis of human gait disorder and metabolic cost estimation. *Arch Appl Mech* 75:569–594
2. Aerospace Medical Research Laboratory (1975) Investigation of inertial properties of the human body. National Highway Traffic Safety Administration
3. Anderson FC, Pandy MG (2001) Dynamic optimization of human walking. *J Biomech Eng* 123:381–390
4. Arnold VI (1989) *Mathematical methods of classical mechanics*. Springer, New York
5. Betsch P (2005) The discrete null space method for the energy consistent integration of constrained mechanical systems. Part I: holonomic constraints. *Comput Methods Appl Mech Eng* 194:5159–5190
6. 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
7. Betsch P, Steinmann P (2001) Constrained integration of rigid body dynamics. *Comput Methods Appl Mech Eng* 191:467–488
8. Betts JT (2009) *Practical methods for optimal control and estimation using nonlinear programming*, 2nd edn. Cambridge University Press, Cambridge
9. Biess A, Liebermann D, Flash T (2007) A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *J Neurosci* 27:13045–13064
10. BostonDynamics (2013) PETMAN. http://www.bostondynamics.com/robot_petman.html

11. Chevallereau C, Aoustin Y (2001) Optimal reference trajectories for walking and running of a bipedal robot. *Robotica* 19:557–569
12. Collins SH, Adamczyk PG, Kuo AD (2009) Dynamic arm swinging in human walking. *Proc R Soc Biol Sci* 276:3679–3688
13. Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelmann E, Thelen DG (2007) Opensim: open-source software to create and analyze dynamics simulation of movements. *IEEE Trans Biomed Eng* 54:1940–1950
14. Djoudi D, Chevallereau C, Aoustin Y (2005) Optimal reference motions for walking of a bipedal robot. In: Proceedings of the 2005 IEEE international conference, vol 1, pp 2002–2007
15. Espiau B, Goswami A (1994) Compass gait revisited. In: IFAC symposium on robot control
16. Felis M, Mombaur K (2013) Modeling and optimization of human walking. In: Mombaur K, Berns K (eds) Modeling simulation and optimization of bipedal walking. Springer, New York
17. Felis M, Mombaur K, Kadone H, Berthoz A (2013) Modeling and identification of emotional aspects of locomotion. *J Comput Sci* 4:255–261
18. François C, Samson C (1996) Energy efficient control of running legged robots. A case study: the planar one-legged hopper. Institut National de Recherche en Informatique et en Automatique
19. Friedmann T, Flash T (2009) Trajectory of the index finger during grasping. *Exp Brain Res* 196:497–509
20. Fujimoto Y (2004) Trajectory generation of biped running robot with minimum energy consumption. In: IEEE international conference on robotics and automation
21. Geyer H, Herr H (2010) A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Trans Neural Syst Rehabil Eng* 18:263–273
22. Geyer H, Seyfarth A, Blickhan R (2003) Positive force feedback in bouncing gaits. *Proc R Soc B: Biol Sci* 270:2173–2183
23. Geyer H, Seyfarth A, Blickhan R (2006) Compliant leg behaviour explains basic dynamics of walking and running. *Proc R Soc B: Biol Sci* 273:2861–2867
24. Gross D, Hauger W, Schröder J, Wall WA (2010) Modeling, Technische Mechanik - Band 3: Kinetik. Springer, Berlin
25. Günther M (1997) Computersimulation zur Synthetisierung des muskulär erzeugten menschlichen Gehens unter Verwendung eines biomechanischen Mehrkörpermodells. Eberhard-Karls-Universität zu Tübingen
26. Hicheur H, Kadone H, Grzes J, Berthoz A (2013) The combined role of motion-related cues and upper body posture for the expression of emotions during human walking. In: Mombaur K, Berns K (eds) Modeling, simulation and optimization of bipedal walking. Springer, New York
27. Koch MW, Leyendecker S (2013) Structure preserving simulation of monopodal jumping. *Arch Mech Eng* LX:127–146
28. Kraft D (1985) On converting optimal control problems into nonlinear programming problems. *Comput Math Program* F15:261–280
29. Leyendecker S (2011) On optimal control simulations for mechanical systems. Technische Universität Kaiserslautern
30. Leyendecker S, Marsden JE, Ortiz M (2008) Variational integrators for constrained dynamical systems. *Zeitschrift für Angewandte Mathematik und Mechanik* 88:677–708
31. Leyendecker S, Ober-Blöbaum S, Marsden JE, Ortiz M (2009) Discrete mechanics and optimal control for constrained systems. *Optim Control Appl Methods* 31:505–528
32. Leyendecker S, Pekarek D, Marsden JE (2013) Structure preserving optimal control of three-dimensional compass gait. In: Mombaur K, Berns K (eds) Modeling, simulation and optimization of bipedal walking. Springer, New York
33. Li Yu, Wang W, Crompton RH, Gunther MM (2001) Free vertical moments and transverse forces in human walking and their role in relation to arm-swing. *J Exp Biol* 204:47–58
34. Lim CL, Jones NB, Spurgeon SK, Scott JJA (2003) Modelling of knee joint muscles during the swing phase of gait - a forward dynamics approach using MATLAB/Simulink. *Simul Model Pract Theory* 11:91–107

35. Luksch T, Berns K (2013) Modeling, simulation and optimization of bipedal walking - modeling and control of dynamically walking bipedal robots. Springer, New York
36. Maas R, Leyendecker S (2013) Biomechanical optimal control of human arm motion. *J Multi-body Dyn.* doi:[10.1177/1464419313488363](https://doi.org/10.1177/1464419313488363)
37. Marsden JE, Ratiu TS (1999) Introduction to mechanics and symmetry. Springer, New York
38. Marsden JE, West M (2001) Discrete mechanics and variational integrators. *Acta Numerica* 10:357–514
39. Mianowski K, Schmitz N, Berns K (2007) Mechatronics of the humanoid robot roman. In: Kozłowski K (ed) Robot motion and control. Springer, London
40. Olver P (1986) Application of Lie groups to differential equations. Springer, New York
41. Pandy MG, Zajac FE, Sim E, Levine WS (1990) An optimal control model for maximum-height human jumping. *J Biomech* 23:1185–1198
42. Pandy MG, Garner BA, Anderson FC (1995) Optimal control of non-ballistic muscular movements: a constraint-based performance criterion for rising from a chair. *J Biomech Eng* 117: 15–26
43. Park J (2008) Synthesis of natural arm swing motion in human bipedal walking. *J Biomech* 41:1417–1426
44. Ren C, Zhao L, Safonova A (2010) Human motion syntheses with optimisation-based graphs. *Comput Graph Forum* 29:1321–1328
45. Rose J, Gamble JG (1994) Human walking. Williams & Wilkins, Baltimore
46. Roussel L, Candas-de Wit C, Goswami A (1998) Generation of energy optimal complete gait cycles for biped robots. In: Proceedings IEEE conference on robotics and automation
47. Safanova A, Hodgins JK (1992) Construction and optimal search of interpolated motion graphs. *Ann Oper Res* 57–373
48. Safanova A, Hodgins JK (2012) Human gait recognition using multisvm classifier. *Int J Sci Res (IJSR)*
49. Simmons G, Demiris Y (2015) Optimal robot arm control using the minimum variance model. *J Robot Syst* 22:677–690
50. Soechting JF, Buneo CA, Herrmann U, Flanders M (1995) Moving effortlessly in three dimensions: does Donders' law apply to arm movement? *J Neurosci* 15:6271–6280
51. Spektrum Akademischer Verlag (1999) Aufrechter Gang. <http://www.spektrum.de/lexikon/biologie/aufrechter-gang/6056>
52. Uno Y, Kawato M, Suzuki R (1989) Formulation and control of the optimal trajectory in human multijoint arm movement. *Biol Cybern* 61:89–101
53. von Stryk O, Bulirsch R (1992) Direct and indirect methods for trajectory optimization. *Ann Oper Res* 37:357–373

Chapter 7

Robotran-YARP Interface: A Framework for Real-Time Controller Developments Based on Multibody Dynamics Simulations

Timothée Habra, Houman Dallali, Alberto Cardellino, Lorenzo Natale, Nikolaos Tsagarakis, Paul Fisettes and Renaud Ronsse

Abstract Multibody dynamics simulation is widely used for prototyping and testing controllers. However, the transfer of controllers initially developed in simulation to real mechatronics platforms requires updating the code in order to interface with physical sensors and actuators. Due to this strong coupling with specific hardware, the controller re-usability is often severely compromised. In the present contribution, we solve this issue by adding a middleware between the controller and the controlled platform (either real or simulated). This framework decouples the controller from the hardware, allows fast controller developments and eases collaborations on large scale projects. Moreover, it offers the possibility to simultaneously control the real and the simulated robot from a unique controller. This paper presents the interface of the Robotran dynamic simulator with the YARP middleware. Robotran leverages symbolic generation of the multibody equations to provide fast and accurate

T. Habra (✉) · P. Fisettes · R. Ronsse
Center for Research in Mechatronics, Institute of Mechanics, Materials,
and Civil Engineering, Université catholique de Louvain (UCL),
Place du Levant 2, 1348 Louvain-la-Neuve, Belgium
e-mail: timothee.habra@uclouvain.be

P. Fisettes
e-mail: paul.fisettes@uclouvain.be

R. Ronsse
e-mail: renaud.ronsse@uclouvain.be

H. Dallali · A. Cardellino · L. Natale · N. Tsagarakis
Advanced Robotics & iCub Facility, Istituto Italiano di Tecnologia (IIT),
Via Morego 30, 16163 Genova, Italy
e-mail: houman.dallali@iit.it

A. Cardellino
e-mail: alberto.cardellino@iit.it

L. Natale
e-mail: lorenzo.natale@iit.it

N. Tsagarakis
e-mail: nikos.tsagarakis@iit.it

simulations of multibody systems. The speed and accuracy of Robotran make it possible to test real-time controllers in a realistic simulation environment. This framework is illustrated with applications using the COMAN and WALK-MAN humanoid robots.

7.1 Introduction

Simulation tools are widely used in prototyping and testing new technologies. By providing a safe and controllable testing environment, they allow fast and cheap software prototyping. In fields such as robotics or vehicle dynamics, mechanical platforms are not always available for controller testing. Indeed, the mechanical design of first prototypes requires time and resources. Moreover, the few platforms usually built up are often subject to repair or maintenance. In addition, running untested controllers turns out to be potentially dangerous for both the operators (injuries) and hardware (damages). In this context, multibody dynamics simulation is a tool being particularly appealing for the development of controllers for complex mechanical systems.

Nevertheless, porting a controller initially developed in simulation to a real device is not straightforward. It typically requires to adapt the format of the input and output signals of the controller to the actual actuators and sensors. Furthermore, in a real setup, the communication and synchronization between the physical sensors, processors, and actuators must be handled. Hence the controller must also interface with the different communication protocols being used. Not only time consuming and error prone, this coupling between the high-level controller and the mechatronic hardware severely impacts the reusability and lifespan of the code. The code stays with a specific mechatronic platform and usually gets obsolete as soon as the hardware changes. This interfacing issue is thus an obstacle to the efficient use of dynamic simulators for controller developments.

Interestingly, an interfacing burden is also faced in robotics when updating a robot hardware or reusing some code initially developed for another robot. Each time a piece of hardware (e.g. sensors, actuators) is changed, the controllers become outdated and need to be modified. This is particularly burdensome in modern projects deploying advanced devices being constantly improved and updated, especially within the research community. To tackle this problem, the use of a so-called “middleware” emerged as a suitable solution to enhance code modularity and to encourage code reuse and collaborations across projects. Applying the principles of a middleware from robotics to dynamic simulators may help to solve the interfacing problem mentioned above.

The principle of a middleware consists in moving all of the code specific to the hardware into an intermediary software lying between the controllers and the robot, namely the middleware (see Fig. 7.1). This software manages the communication between the controllers and the actual device. It provides the controllers with an interface to read sensors and drive actuators. Thus, the controllers become totally

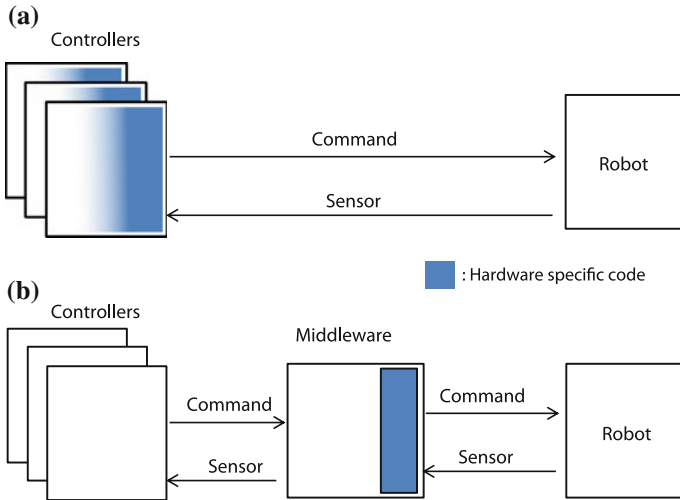


Fig. 7.1 Blue rectangles represent the piece of code being specific to the hardware. **a** Controllers directly commanding the robot, with their hardware-specific code. **b** Controllers interfaced with a middleware and having no hardware-specific code

independent of the hardware. People developing controllers can then focus on their algorithm and are set free from the interfacing with the device. The lifespan of the code is then extended as it does not need to be modified when some pieces of hardware are updated or changed. By decoupling the controllers from the robot hardware, it further allows a unique controller to work with different pieces of hardware (e.g. joint encoders) as long as they are properly interfaced with the middleware.

This approach can also be applied with a multibody simulator. Indeed, from a controller perspective, a multibody simulator can be seen as a collection of virtual hardware (actuators and sensors). Therefore, as for any physical device, the virtual hardware of a simulator can be interfaced with a middleware that takes care of matching the relevant input/output signals. Then, without changing a single line of code, the very same controller can work with the simulator and with the real robot.

Over the years, several middleware frameworks were developed. Generic middleware like CORBA,¹ Ice² or ØMQ³ provide complete communication backbones and are widely used to communicate through the internet network. However, they are rarely employed in robotics. Firstly, they are not easy to adopt by non-experts. Secondly, they lack specific components for robotics such as interfaces for sensors and actuators. For that reason, middleware being specific to robotics appeared, such as Player [5], ROS [29], Orocos [3], YARP [25], Orca [24], OpenRDK [4], Mira [10] or LCM [18] to mention just a few.

¹<http://www.corba.org/>.

²<http://www.zeroc.com/ice.html>.

³<http://www.zeromq.org/>.

Player [5] is a *hardware abstraction layer* for robotic devices. It implements drivers for a large collection of sensors and robots. The Stage simulator is also interfaced with Player and offers to simulate a population of mobile robots in 2D [15]. It is not intended to simulate multibody dynamics but rather to test planning algorithms.

ROS [29] can be seen as the evolution of the Player/Stage framework. It was launched in 2007 and has gained a wide audience ever since. It is interfaced with the Gazebo simulator [21] which was initiated as a 3D complement to Stage. Not only a middleware, ROS also provides a rich array of toolboxes and packages for robotics developed among a large community. ROS defines some standard message formats for sensors and provides interfaces for controllers. The Player drivers can be used in ROS.

Orocos middleware [3] builds its communication around CORBA. It is popular in robotics for its kinematics and dynamics library (KDL) that can be easily integrated into controllers. Orocos direct dynamics module in addition with a time integrator can be used to create a custom simulator [12]. However, contacts with the environment are not included in this scheme. Orocos can also connect to the Gazebo simulator through ROS.

YARP middleware [25] is mainly used in humanoid robotics, and most notably for the iCub platform [26]. For such a kind of robot with many actuators and sensors, YARP facilitates the code modularity by abstracting the hardware through interfaces. It can be extended with device drivers to be compatible with a variety of sensors and actuators. Originally, a dynamic simulator based on Open Dynamics Engine (ODE) was developed for the iCub [32]. However, it was not generic enough to simulate other robots. Therefore, a plugin to use YARP with Gazebo was also developed [17].

For more details about the different middleware used in robotics, the interested reader can refer to the following surveys [22, 27].

Currently, the multibody research community is not sufficiently connected with the robotics research community. In this regard, we believe that the adoption of middleware in the multibody domain could facilitate collaborations. This would be beneficial for both robotics and multibody researchers. On the one hand, roboticists would gain access to the state-of-the-art in multibody dynamics simulators. Indeed, the middleware interface would guarantee an effortless transfer of their controllers to new simulators. A strong current trend in robotics is to use simulators being initially developed for video games, like ODE and Bullet [19]. These simulators are primarily targeting fast computation with results being visually-plausible rather than physically correct. Therefore, adoption of simulators from the multibody community should improve the accuracy of the simulations. On the other hand, the multibody community could improve the code modularity and reusability, and more generally, enhance collaborations using software development techniques coming from robotics.

This work presents the interfacing of the Robotran multibody dynamics simulator [8] with the YARP middleware [25]. Similar interfaces between middleware and simulators already exist: for example YARP with Gazebo [17] and ROS

with Gazebo.⁴ The Gazebo simulator is attractive and widely used in the robotics community because it allows to model complex world and vision sensors. Moreover, Gazebo and ROS communities are large, active, and partly overlap. However, none of these existing solutions is satisfying enough to get accurate simulation in realistic scenarios while remaining faster than real-time. This demanding tradeoff in terms of computational speed and simulation accuracy is an incentive for using Robotran. Indeed, thanks to its symbolic formalism, Robotran can generate for any given rigid multibody model, its corresponding equations of motion in a form specifically tailored for fast computation. By removing the unnecessary arithmetic operations of a model, Robotran yields an optimal form of the kinematic and dynamic equations and of the associated routines [8].

Importantly, the user should not lose the flexibility offered by the middleware by being forced to use a single simulation environment. Therefore, a controller developed with the Robotran-YARP interface remains totally compatible with the Gazebo-YARP interface. Indeed, since the Gazebo-YARP and Robotran-YARP plugins implement all the same drivers, the same controllers can also work with both. This is precisely the objective of a middleware. Also, YARP has the capability to interoperate with ROS. Therefore, our interface is also compatible with software being developed by the ROS community.

The presented approach remains general and can be followed for other multibody simulators and other middleware. Robotran was selected as the dynamic simulator for its speed and accuracy provided by its symbolic formalism. YARP was chosen for its openness and interoperability (e.g. fully compatible with ROS). The corresponding code is open source and publicly available.⁵

This chapter is structured as follows. Section 7.2 outlines the main features and implementation of the YARP middleware. Section 7.3 presents the drivers implemented to interface Robotran with YARP. Then, Sect. 7.4 details some applications enabled by the proposed framework. To illustrate how the proposed framework can be used for controllers of complex mechatronic systems, examples with the COMAN [33] and WALK-MAN [11] humanoid robots are given. Finally, Sect. 7.5 gives a brief conclusion.

7.2 YARP Middleware

A major problem in robotics is that controllers can quickly get entangled with the platform they are running on and the devices they are controlling. Moreover, many modules are typically used to control a robot. Managing the inter-module communication can quickly become intractable and time consuming. In that context, the YARP robotics framework has been developed to help reduce the time spent doing infrastructure-level programming rather than actual research [13, 25].

⁴http://www.wiki.ros.org/gazebo_ros_pkgs.

⁵In public repositories <https://www.gitlab.robotran.be/walkman>.

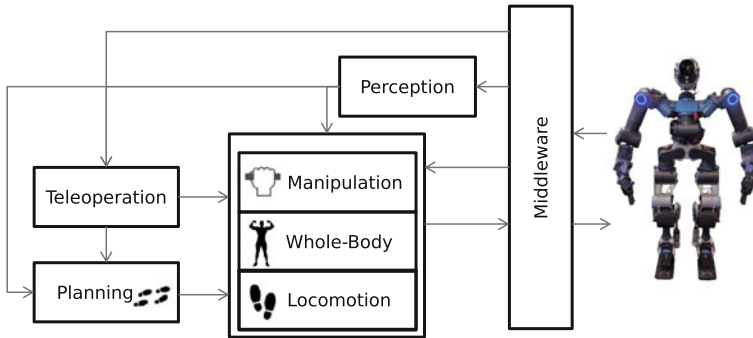


Fig. 7.2 Overview of the control framework of the WALK-MAN robot. It represents a typical example of the modules being used to control the humanoid robot performing the tasks of the DARPA robotics challenge

7.2.1 Motivation

A typical example of a middleware utility is raised in the WALK-MAN project, aiming at developing a humanoid robot able to support rescue teams by performing activities in the most hazardous disaster zones⁶ (e.g. Fukushima nuclear plant). Example of tasks to be achieved are those proposed at the recent DARPA Robotics Challenge, such as valve turning, stairs climbing or debris removal.⁷ These tasks require integration of multiple modules of perception, locomotion and manipulation. An overview of this high level control framework is presented in Fig. 7.2.

This type of collaborative research project requires various modules to be developed and shared among multiple research groups located in different countries (in this case Italy, Switzerland, Germany and Belgium). During the first months of the project, the WALK-MAN robot was designed. Consequently, the first software developments were performed in simulation or by using other humanoids robots (e.g. the COMAN [33]). Last but not least, some controllers and software tools previously developed in other projects were expected to be reused.

Without using a middleware, all of the software developments and integration achieved in a time period of 18 months would have been impossible.

7.2.2 Middleware Principle

The key idea of a middleware is to extract the hardware-controller interface out of the controller code itself. As shown in Fig. 7.1, all of the hardware dependent code

⁶See <http://www.walk-man.eu/>.

⁷<http://www.theroboticschallenge.org/>.

is handled by the middleware. The different controllers are then totally “agnostic” with respect to the actual hardware being used.

The hardware dependent codes, known as the *device drivers*, manage the direct operation of the hardware devices. It can be for example the code requesting the joint position measured by an encoder communicating through an I2C bus. It can also be the code sending a desired voltage command through a CAN bus to the power electronics driving a DC motor. Thus, for every actuator or sensor communicating with the controller, a driver must be implemented. All of the drivers of a specific platform are gathered in the middleware.

In order to keep the controllers flexible to hardware changes, *device interfaces* are used between the user controllers and the middleware. They consist of hardware abstraction layers for families of devices. The idea is to provide a unique interface for all the devices having the same functions, regardless of the technology they rely on (see Fig. 7.3). The user control algorithms are blind with respect to the actual devices being used on the robot. The only relevant information they depend on is the type of interface being available on the mechatronic platform. For example, the encoder interface can be used for any sensor providing a measurement of the joint position. If an upgrade of the platform requires to replace optical encoders by magnetic encoders, the controllers are not affected. Programmers can write the higher-level application code independently of the specific hardware of a particular platform in a particular configuration.

This abstraction through device interfaces is actually borrowed from classic computer operating systems. For example, a computer software can use an interface for the clicking devices. The interface makes it unnecessary to re-install all computer applications after plugging in a mouse or when using a new touch screen. Instead, it is simply required to have the correct drivers for these devices while they have to comply with the same interface.

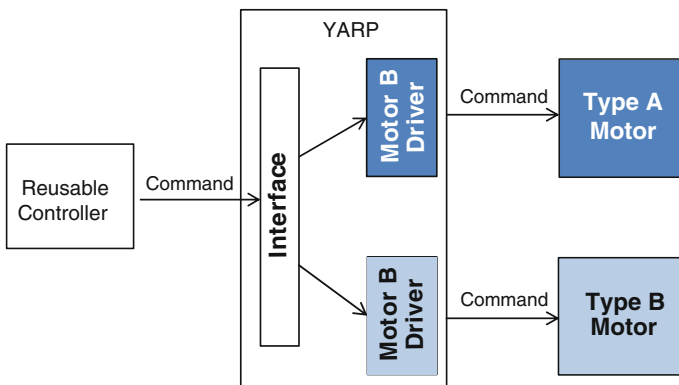


Fig. 7.3 Different devices of the same family, *Type A Motor* and *Type B Motor* are controlled by the same controller. Each motor has its own driver but both drivers implement the same interface

7.2.3 Implementation

Practically in YARP, the device interfaces consist of Application Programming Interface (API) used to control actuators and read sensors data. They are implemented in C++ as abstract base classes with a set of virtual methods. So, each type of sensor (e.g. joint position encoders) is associated to a set of interfaces specifying the collection of functions available to access the data (e.g. *getEncoder()*). Similarly, each type of actuator (e.g. a joint DC motor) is associated to interfaces specifying the functions to control it (e.g. *positionMove(ref)*). For more details about YARP interfaces, see [13].

Following the object oriented programming paradigm, device drivers are children of the device interface classes. So for each device interfaced with YARP, a *driver* should implement the corresponding *interface* classes. Drivers of different devices belonging to the same family can implement the same interface. This allows different types of encoders to implement the same *encoder interface* and thus to be compatible with the same controller. As written above, this architecture makes the controller be device agnostic: it knows the interface functions, but does not need to know which specific device implements them. The controller should only fulfill the requirements regarding the type and format of the data it gets or needs to send. So each device can implement interfaces when appropriate, with the requirement of being compliant with the specifications, e.g. regarding the unit (degrees for angular positions, Newton for forces, etc.). This guarantees that the controller can work with any platform implementing its interfaces, as illustrated in Fig. 7.3.

In addition, for each interface, YARP implements generic network proxy devices allowing the remote execution of the same code. Therefore, different controller pieces can run on different machines. Thus, controllers on a user laptop can be interchangeably connected to the physical mechatronic platform or to a simulator with no necessity to recompile the program. This offers for instance to rely on the specific advantages of specialized computers for specific computations (e.g. for vision or intensive computation).

7.3 Robotran-YARP Drivers

Since a simulator can be seen as a virtual robot, it is possible to implement drivers for the simulated devices. This was performed for the Robotran simulator, as described in this section. This implementation allows to test controllers in simulation and then to port them to the real robot without changing a single line of code. Figure 7.4 illustrates an example of this structure with the COMAN humanoid robot.

Robotran is particularly tailored to fast and accurate multibody dynamics simulations [30]. To challenge the capacity of Robotran to produce fast and accurate simulations, the proposed interface aims at testing robotic platforms and controllers involving large dynamical contributions (e.g. locomotion, whole-body control).

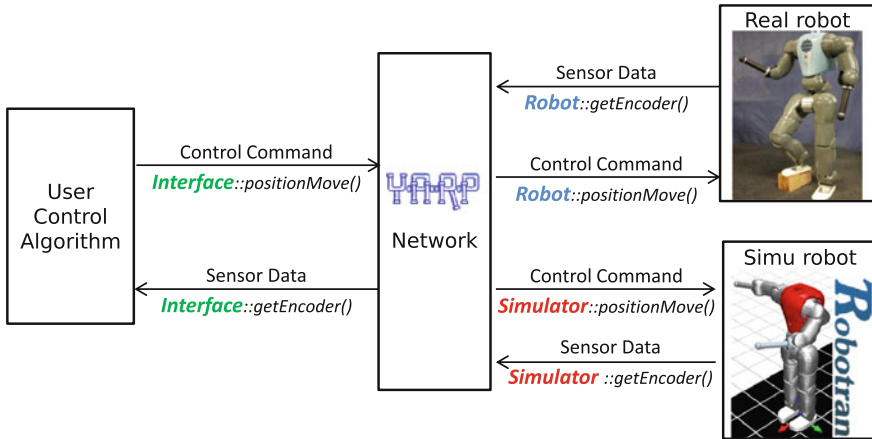


Fig. 7.4 Robotran-YARP control framework. The user control algorithm calls the interfaces functions without knowing what device is behind. The real robot and the one simulated with Robotran, have their own drivers implementing these interfaces. The very same controller can then be used with both the real and the simulated robot

Consequently, the drivers for actuators and for the sensors measuring dynamics (i.e. torque sensors rather than cameras) are implemented in the Robotran environment.

7.3.1 Control Board Driver

On a real robot, control boards are electronic boards supervising the joints actuation. A control board often embeds a joint control algorithm implemented at the firmware level. Different control modes are usually available (e.g. position or torque control). As displayed in Fig. 7.5, this low-level control algorithm drives the actuator and receives a feedback from the joint encoder. If available, it can also acquire a torque

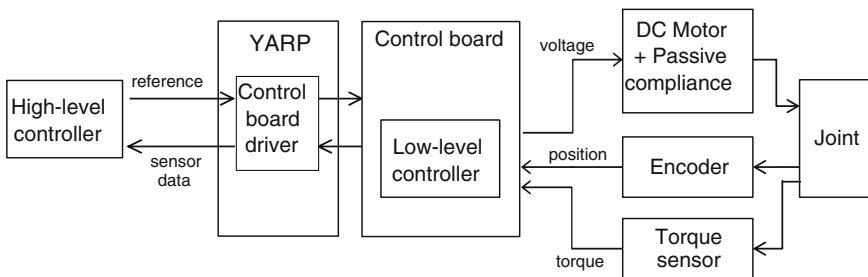


Fig. 7.5 The control board is a key bridge between the high-level controller and the actual joints. It contains a low-level controller directly controlling the actuators

feedback from a joint torque sensor. These low-level joint controllers can usually be decoupled from the high-level control algorithms. While the high-level controllers (e.g. locomotion algorithm) run on a central on-board PC, the low-level controllers (e.g. torque control) directly run on the electronic boards. Typically, high-level algorithms do not directly drive the motors but rather send references (position, velocity, torque, etc.) to control boards that will, in turn, drive the motors (see Fig. 7.5). Therefore, a control board driver receives reference commands from the user control algorithm and translates it into commands being specific to the low-level joints controller.

The control board driver is thus in charge of several devices and implements many device interfaces. The main interfaces implemented for the Robotran control board driver are:

- *IEncoders*: returns the joints position, velocity and acceleration.
- *IOpenLoopControl*: sends a voltage command to the DC motors in open loop (without feedback control). This mode thus bypasses the low-level controller mentioned above, allowing feedforward control.
- *IPositionControl*: controls the joint position. It can implement either a PID control (stiff mode) or a joint impedance control (compliant mode).
- *IVelocityControl*: controls the joint velocity. It can implement either a PID control (stiff mode) or a joint impedance control (compliant mode).
- *ITorqueControl*: controls and measures the joint torque.
- *IControlMode2* and *IInteractionMode*: select respectively the control mode (position, velocity, torque, open loop) and the interaction mode (stiff or compliant).
- *IPidControl* and *IImpedanceControl*: control respectively the low-level PID controller gains and the virtual impedance stiffness and damping.
- *IAmplifierControl*: feeds back the electrical current in the motors.

In order to make the simulation environment compatible to the real hardware, two add-ons were implemented in the Robotran simulation suite (and more particularly the devices presented in Fig. 7.5). These add-ons are generic in the sense that they can be reused for different Robotran multibody models by adapting some configuration files.

The first add-on consists of joints controllers mimicking the firmware low-level controllers of the real robot. This allows to specify for each joint the desired type of control (position, torque, impedance and open-loop). The PID gains can also be modified through this module.

The second add-on consists in a compliant actuator model, which is particularly useful to simulate compliant robots like COMAN [33] and WALK-MAN [6]. It provides the constitutive laws of a DC motor and the mechanical model of a series elastic actuator (SEA). This model is combined with the multibody dynamics through the Robotran *user derivatives* feature. The latter allows integration of the additional differential equations of the DC-motor and the joint compliance together with the set of multibody dynamic equations. All the details can be found in [7, 34].

Based on these add-ons, the Robotran Control Board Driver is implemented. At each simulation time step, joints position, velocity and torque are read by the driver

and sent to YARP. The desired joints position or torque are then received from YARP and sent to the Robotran low-level controller.

7.3.2 *Force-Torque Sensor Driver*

A 6-axis force-torque sensors measure the wrenches acting on a body, i.e. the 3 force components (F_x , F_y , F_z) and the 3 torque components (T_x , T_y , T_z) expressed in a body fixed frame.

On a real robot, this type of sensor is typically made of strain gauges capturing small deformations. In Robotran, this can be simulated by means 6 locked joints at the sensor location. The Lagrange multipliers computed by Robotran provides the force or torque being required to keep these joints locked [30]. Assuming that the body deformation due to the strain gauge stretching are negligible, this technique is a good approximation for modelling this sensor.

This driver thus implements the *IAnalogSensor* interface which outputs 6 values representing the forces and torques components.

7.3.3 *IMU Sensor Driver*

An Inertial Measurement Unit (IMU) measures the orientation, velocity and acceleration of a body.

This physical sensor usually combines accelerometers and gyroscopes [9].

In the simulation environment, the same information can be retrieved using the Robotran forward kinematics module [30]. Placing a Robotran sensor at the same location as the IMU sensor provides position, orientation, velocity and acceleration of this point in space. Note that Robotran provides the body orientation by means of a rotation matrix. It must be converted into Euler angles to comply with the YARP format.

In summary, this driver implements an *IAnalogSensor* interface which outputs the orientation, the angular velocity and the linear acceleration of a body.

7.3.4 *Clock Driver*

This driver gets the simulation time in order to synchronize the controllers with the simulation environment. Such a driver is thus specific to the simulation environment, and does not exist in the physical counterpart. Indeed, a simulated time clock is not always constant. It might run faster or slower than real time. If the controllers and the simulator are not synchronized, they will display unrealistic behaviors. All the interconnected systems should thus follow the same clock managed by the simulator

clock driver. This driver distributes the clock through the network to all the involved machines and applications. Note that this does not prevent fast simulators like Robotran to run tests of real-time controllers even faster than real time. Indeed, running the simulator clock faster than real-time is possible, as long as all interconnected systems stay synchronized.

Moreover, the Robotran library provides a tool to control the simulation time. The clock can be artificially slowed down in order to carefully observe the device behavior during highly dynamical events, for example, an impact with the ground or an object. Moreover, this central clock even offers to rewind a simulation backward in time, a feature that proves to be very useful when debugging.

7.4 Applications

The Robotran-YARP framework offers a wide range of applications. Its standardized interface enhances the code reusability and eases the transfer of a controller from a simulation environment to an experimental setup. Moreover, it can be used to simultaneously run a controller on the simulator and on the experimental setup. Finally, the use of a middleware can also facilitate the benchmarking of multibody simulation tools. These different examples are detailed in the sections hereunder.

7.4.1 Code Reusability

YARP clearly specifies the interface of the control modules. To (re)use a control module, the only request is to connect it to the correct inputs-outputs through the corresponding YARP communication ports. This “plug and play” architecture eases the software reuse across projects.

As an example of module reusability, Fig. 7.6 shows the *Robot Motor Gui* module controlling the simulated COMAN. This YARP module implements a graphical user interface to interact with the control boards. It offers to test different low-level joint controllers. It is interesting to note that, at the time this module was written, no interface between YARP and Robotran existed. Nevertheless, as the module was designed to be compliant with YARP, it could be reused in Robotran simulated platform without any modification. Moreover, this module was initially developed for another robot, namely the iCub [26] and was imported here without changing a single line of code, despite a very different hardware. This illustrates the potential of code reusing and cross-projects collaboration brought by our framework.

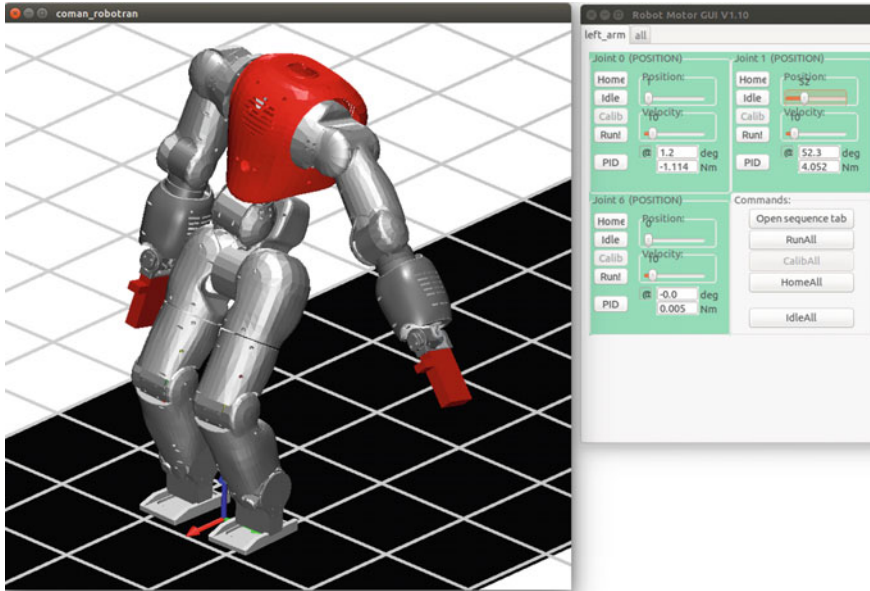


Fig. 7.6 The COMAN humanoid simulated in Robotran is controlled by a classical YARP-module, the “Robot Motor GUT” through the control board interface

7.4.2 Code Transfer

Our framework further allows to seamlessly transfer a module from simulation to real hardware and *vice versa*. Thanks to the middleware decoupling, the controllers do not need to be recompiled when switching from one environment to the other (simulated or physical). Furthermore, this seamless transfer to simulation makes possible to write regression tests in order to verify that patches or new developments do not modify the expected behavior of the robot software. Controller testing can thus be automated, as prescribed by the software continuous integration methodology [14].

As represented in Fig. 7.7, a squatting task was run on both Robotran and the real WALK-MAN robot in order to compare the simulated model of WALK-MAN with experiments. The position, torque and the 6 DoF force torque sensor data were logged for comparison. The data is presented for 30s where the robot squats down by 30 cm and returns to its initial posture every 7 s, completing a full cycle. In terms of position tracking, once the low-level PID controllers are properly tuned, a suitable tracking performance is obtained. Regarding the knee torques, there are some differences between the simulation data and the experimental results. The experimental data is more oscillatory. This is due to parasitic movements in the other joints, while this is better controlled in the simulation. Also the average torque profile is slightly higher in the experiment. This is due to unmodelled shock absorbing plastic covering the robot body and to electrical wires and electronic boards adding weight to the actual

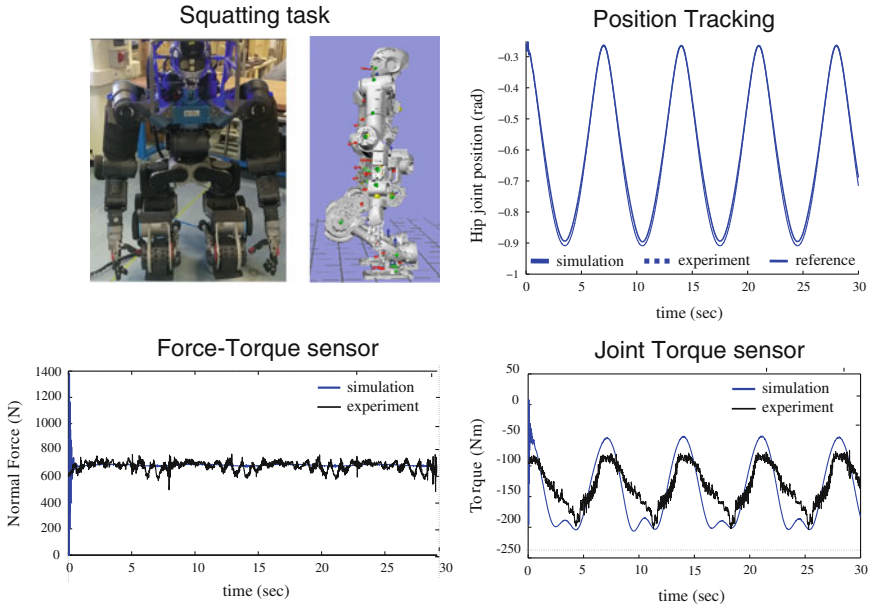


Fig. 7.7 Squatting task performed by a simulated and real WALK-MAN robot. The leg joints are controlled in position to move the robot from the standing up to the squatting posture. *Top left* figures show the real and simulated robots in the squat posture. *Top right* figure represents the evolution of the position tracking of the Hip joint. *Bottom left* figure represents the measurement of the vertical force in the foot sensor. *Bottom right* figure compares the experimental and the simulated torque in the knee joint

robot, while this extra weight was not initially captured in the model. This difference is indeed reduced after adjusting the modelled torso weight, as shown in the figure. Note that a better prediction should likely require an extended system identification for refining the model.

7.4.3 Parallel Execution

On top of that, the simulation can also run in parallel to the real robot (see Fig. 7.8). Indeed, the modular interface of the middleware offers to simultaneously control a real robot and a simulated one (the commands being duplicated).

This offers the possibility to add an *internal model* in the control scheme, i.e. a framework with a strongly bio-inspired ground [20]. As sketched in Fig. 7.9, an internal model consists in a dynamic simulator receiving a copy of the commands sent to the real platform. It can then predict the state of the robot. Such a model provides a particularly useful tool to perform *state estimation*, *model identification* and *predictive control* [2, 16, 23, 31].

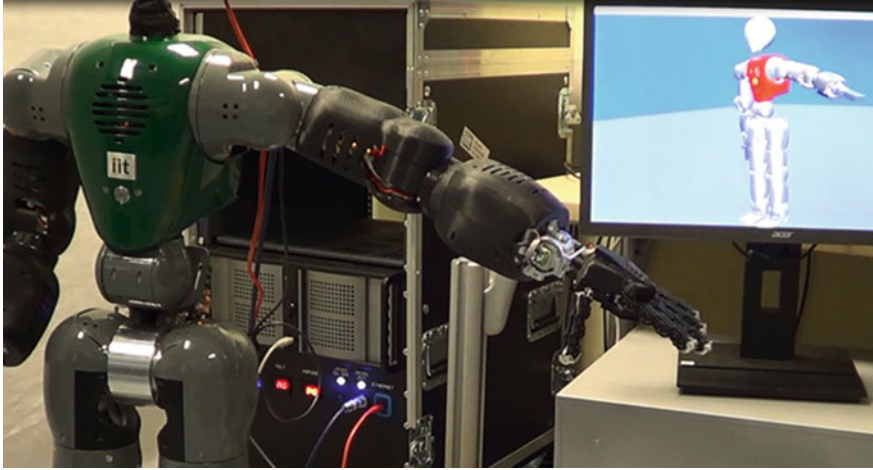


Fig. 7.8 Real and simulated COMAN humanoid controlled in parallel by a unique controller

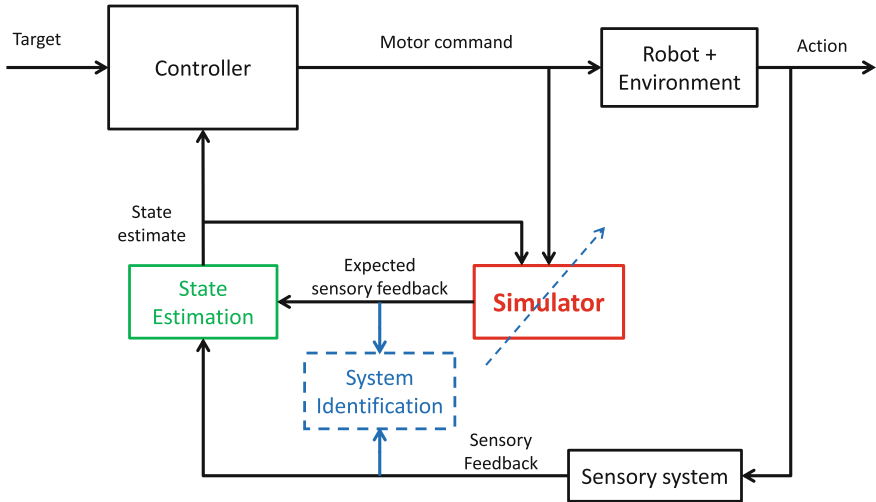


Fig. 7.9 Control scheme using a forward internal model (in red, in our case represented by the simulator block). The internal model receives a copy of the command and predicts the expected sensory output. This information can then be used for system identification and state estimation (Adapted from [31])

Among others, we plan to use this tool in the next future to support robotic teleoperation with low bandwidth and noisy communication channels between the robot and its operator.

7.4.4 Benchmarking

Finally, our framework can also be used to support the benchmarking of multibody simulators. Indeed, in the same way a controller can be transferred from a simulated to a real robot, a benchmark test could be transferred from a simulation environment to another. Provided that the simulators are interfaced with the same middleware, any benchmarking test could be performed with no adaptation of the controller code. Following this approach, it is guaranteed that the benchmark is performed in the same conditions for all the simulators under test. This is similar to the benchmarking in [1, 28] using a physics abstraction framework to switch from one simulator to another.

7.5 Conclusion

In conclusion, a clean interface between Robotran and YARP was developed in a generic and open source library. The use of a middleware interfaced with a multibody simulator offers many advantages for software development.

Firstly, the YARP middleware provides all the tools being necessary to create modular projects. It decouples the high level controllers from the controlled platform (real or simulated). This modularity eases the creation, maintenance and reuse of the software modules. This naturally supports collaborations and enhances the code durability.

Secondly, the integration of a dynamic simulator provides a powerful testing environment. It allows to perform early and regular tests in a safe and cheap environment. It is important to recall that, in the proposed framework, the code tested in the simulation environment is exactly the same as the one ported to the real platform. Interestingly, the bugs can further be replicated! This testing capability is key to produce high quality software modules.

Finally, the adoption of middleware among multibody research groups could tighten the link with robotics researchers: the middleware being the glue between the multibody simulation environments of the ones and the high level controllers of the others.

The modularity and testing capabilities enabled by our framework were successfully implemented with the WALK-MAN and COMAN robots. Fast software prototyping could thus be achieved thanks to the reusability brought by the YARP interface. Moreover, the transfer of a controller tested in a simulation to the real platform is now straightforward: the exact same code can run in simulation or on a real experimental setup. More advanced applications such as the parallel execution of the same controller in both the simulation and in the actual robot are also possible. This allows to perform state estimation and online system identification, i.e. two features being critical for robots working in unknown environments. We also envisioned that this framework could be used for the benchmarking of multibody simulators.

Future works will consist of further extending the collection of robots available to other platforms, like the iCub humanoid robot [26]. We also plan to use it to perform online system identification of the robots and its environment. Drivers for other sensors might also be implemented (tactile sensors, camera, etc.).

The proposed framework allows to improve and foster software developments among large collaborative projects such as the WALK-MAN project. We hope it can benefit to other research consortia. Therefore, we released the open source code of the Robotran-YARP plugin together with the simulators of the COMAN and WALK-MAN humanoids in a public repository (<https://www.gitlab.robotran.be/walkman>).

Acknowledgments This work was supported by the European Community's 7th Framework Programme (FP7-ICT-2013-10) under Grant 611832 (WALKMAN collaborative project) and by the Belgian F.R.S.-FNRS (Crédit aux Chercheurs 6809010 awarded to Renaud Ronsse). We are grateful to Alexandra Zobova and Nicolas Van der Noot for their support in the development of the simulator.

References

1. Boeing A, Brunl T (2007) Evaluation of real-time physics simulation systems. In: Proceedings of the 5th international conference on computer graphics and interactive techniques in Australia and Southeast Asia, ACM, pp 281–288
2. Bongard J, Zykov V, Lipson H (2006) Resilient machines through continuous self-modeling. *Science* 314(5802):1118–1121
3. Bruyninckx H (2001) Open robot control software: the orocos project. In: Proceedings of the IEEE international conference robotics and automation (ICRA), vol 3, pp 2523–252
4. Calisi D, Censi A, Iocchi L, Nardi D (2012) Design choices for modular and flexible robotic software development: the OpenRDK viewpoint. *J Softw Eng Robot* 3(1):13–27
5. Collett TH, MacDonald BA, Gerkey BP (2005) Player 2.0: toward a practical robot programming framework. In: Proceedings of the Australasian conference on robotics and automation (ACRA 2005), p 145
6. Dallali H, Mosadeghzad M, Medrano-Cerda G, Loc VG, Tsagarakis N, Caldwell D, Gesino M (2013) Designing a high performance humanoid robot based on dynamic simulation. In: European modelling symposium (EMS), 2013, IEEE, pp 359–364
7. Dallali H, Mosadeghzad M, Medrano-Cerda GA, Docquier N, Kormushev P, Tsagarakis N, Li Z, Caldwell D (2013) Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach. In: 2013 IEEE international conference on mechatronics (ICM), IEEE, pp 598–603
8. Docquier N, Poncelet A, Fisette P (2013) Others: robotran: a powerful symbolic generator of multibody models. *Mech Sci* 4:199–219
9. Dudek G, Jenkin M (2008) Inertial sensors, GPS, and odometry. In: Springer handbook of robotics, Springer, Berlin, pp 477–490
10. Einhorn E, Langner T, Stricker R, Martin C, Gross HM (2012) Mira-middleware for robotic applications. In: 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 2591–2598. IEEE (2012)
11. Eu project whole body adaptive locomotion and manipulation (walkman). <http://walk-man.eu/>
12. Ferraguti F, Golinelli N, Secchi C, Preda N, Bonfe M (2013) A component-based software architecture for control and simulation of robotic manipulators. In: 2013 IEEE 18th conference on emerging technologies and factory automation (ETFA), IEEE, pp 1–5
13. Fitzpatrick P, Metta G, Natale L (2008) Towards long-lived robot genes. *Robot Auton Syst* 56(1):29–45

14. Fowler M, Foemmel M (2006) Continuous integration. Thought-Works. <http://www.thoughtworks.com/ContinuousIntegration.pdf>
15. Gerkey B, Vaughan RT, Howard A (2003) The player/stage project: tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th international conference on advanced robotics, vol 1, pp 317–323
16. Haruno M, Wolpert DM, Kawato M (2001) MOSAIC model for sensorimotor learning and control. *Neural Comput* 13(10):2201–2220. doi:[10.1162/089976601750541778](https://doi.org/10.1162/089976601750541778)
17. Hoffman EM, Traversaro S, Rocchi A, Ferrati M, Settini A, Romano F, Natale L, Bicchi A, Nori F, Tsagarakis NG, (2014) Yarp based plugins for gazebo simulator. In: Proceedings of the first international workshop modelling and simulation for autonomous systems, MESAS 2014, vol 8906. Rome, 5–6 May 2014, Revised Selected Papers, Springer, p 333
18. Huang AS, Olson E, Moore DC (2010) LCM: Lightweight communications and marshalling. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 4057–4062
19. Ivaldi S, Peters J, Padois V, Nori F (2014) Tools for simulating humanoid robot dynamics: a survey based on user feedback. In: 2014 14th IEEE-RAS international conference on humanoid robots (Humanoids), IEEE, pp 842–849
20. Kawato M (1999) Internal models for motor control and trajectory planning. *Curr Opin Neurobiol* 9(6):718–727. doi:[10.1016/S0959-4388\(99\)00028-8](https://doi.org/10.1016/S0959-4388(99)00028-8)
21. Koenig N, Howard A (2004) Design and use paradigms for gazebo, an open-source multi-robot simulator. In: Proceedings of the 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS 2004), vol 3, IEEE, pp 2149–2154
22. Kramer J, Scheutz M (2007) Development environments for autonomous mobile robots: a survey. *Auton Robot* 22(2):101–132
23. Li Q, Poo AN, Lim CM (1996) Internal model structure in the control of robot manipulators. *Mechatronics* 6(5):571–590
24. Makarenko A, Brooks A, Kaupp T (2006) Orca: components for robotics. In: 2006 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 163–168 (2006)
25. Metta G, Fitzpatrick P, Natale L (2006) YARP: yet another robot platform. *Int J Adv Robot Syst* 3(1):43–48
26. Metta G, Sandini G, Vernon D, Natale L, Nori F (2008) The iCub humanoid robot: an open platform for research in embodied cognition. In: Proceedings of the 8th workshop on performance metrics for intelligent systems, ACM, pp 50–56
27. Mohamed N, Al-Jaroodi J, Jawhar I (2008) Middleware for robotics: a survey. In: 2008 IEEE conference on robotics automation and mechatronics, IEEE, pp 736–742
28. Peters S, Hsu J (2015) Simple benchmarks for speed and accuracy of rigid body dynamic simulators. In: Proceedings of the ECCOMAS thematic conference multibody dynamics 2015
29. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY (2009) ROS: an open-source robot operating system. In: Proceedings of the ICRA workshop on open source software, vol 3, p 5
30. Samin JC, Fiset P (2003) Symbolic modeling of multibody systems, vol 112. Springer Science and Business Media
31. Shadmehr R, Krakauer JW (2008) A computational neuroanatomy for motor control. *Exp Brain Res* 185(3):359–381
32. Tikhonoff V, Cangelosi A, Fitzpatrick P, Metta G, Natale L, Nori F (2008) An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator. In: Proceedings of the 8th workshop on performance metrics for intelligent systems, ACM, pp 57–61
33. Tsagarakis NG, Morfey S, Cerda GM, Zhibin L, Caldwell DG (2013) Compliant humanoid coman: optimal joint stiffness tuning for modal frequency control. In: Proceedings of the 2013 IEEE International Conference on robotics and automation (ICRA), IEEE, pp 673–678
34. Zbova A, Habra T, Van der Noot N, Dallali H, Tsagarakis N, Fiset P, Ronsse R (2015) Others: multi-physics modelling of a compliant humanoid robot. In: Proceedings of the ECCOMAS thematic conference multibody dynamics 2015

Chapter 8

Wheel-Ground Modeling in Planetary Exploration: From Unified Simulation Frameworks Towards Heterogeneous, Multi-tier Wheel Ground Contact Simulation

Roy Lichtenheldt, Stefan Barthelmes, Fabian Buse
and Matthias Hellerer

Abstract Today's growing scientific interest in extraterrestrial bodies increases the necessity of extended mobility on these objects. Thus, planetary exploration systems are facing new challenges in terms of mission planning as well as obstacle and soil traversability. In order to fit the tight schedules of space missions and to cover a large variety of environmental conditions, experimental test setups are complemented by numerical simulation models used as virtual prototypes. In this context we present an integrated simulation environment which allows for using different available contact models, ranging from simple but real-time capable approximations based on rigid-body modeling techniques up to very accurate solutions based on Discrete Element Method (DEM). The models are explained and classified for their applications. For this work, a one-point Bekker based approach (BCM) and the so-called Soil Contact Model (SCM), which is a multi-point extension of the Bekker–Wong method taking soil deformation into account, are used for further analysis. These two contact models are applied for homogeneous simulations with only one type of contact model for all wheels as well as for a heterogeneous multi-tiered simulation with different contact models for the wheels. It will be shown that the multi-tiered approach enhances the simulation result accuracy compared to the results of a homogeneous model with a low level of detail while speeding up the simulation in comparison to a homogeneous higher-tier model.

Keywords Wheel-ground interaction · Terramechanics · Soil contact · Multi-tier modeling · Planetary exploration

R. Lichtenheldt (✉) · S. Barthelmes · F. Buse · M. Hellerer
Deutsches Zentrum für Luft- und Raumfahrt/German Aerospace Center (DLR)
Robotics and Mechatronics Center Institute of System Dynamics and Control,
Münchener Straße 20, 82234 Weßling, Germany
e-mail: Roy.Lichtenheldt@DLR.de

S. Barthelmes
e-mail: Stefan.Barthelmes@DLR.de

8.1 Introduction

In order to further understand the formation of planets and our solar system, planetary science requires extended mobility for the exploration of extraterrestrial bodies. Therefore, the locomotion sub-systems enabling planetary exploration are facing new challenges in terms of durability and mission planning as well as obstacle and soil traversability. Testing in the actual environmental conditions is often very expensive and time-consuming or not even possible. Additionally, environmental conditions on the site of operation are often uncertain and not well-known beforehand. To cover this large variety of parameters for optimizing the locomotion sub-system and fit the tight schedules of space missions, experimental test setups are more and more complemented by numerical simulation models.

In this context we present the DLR Rover Simulation Toolkit (RST) which is an integrated simulation environment dedicated to the design of planetary rovers. For the wheel-ground contact, the RST uses an in-house developed Contact Dynamics Library (CDL) with a unified interface and modular design. This enables a straight forward implementation of rover locomotion system models including seamless switching between contact models with very different level of detail. The contact models range from simple but real-time capable approximations based on rigid-body modeling techniques via penetration and soil deformation approaches to very accurate but slow particle-based methods. Having these different techniques available in one environment allows us to directly compare results of different tier models amongst each other. Furthermore, running different contact models within one locomotion system model, which we call *heterogeneous simulations*, is exploited in this work. Our in-house Soil Contact Model (SCM), which was previously verified in [1], is the highest tier model that is still computationally efficient enough to use it in multi-body dynamics simulations. It is thus used as reference for the comparison of homogeneous lower-tier and multi-tiered heterogeneous wheel ground contacts in this work. A first feasibility study of this method was carried out in [2] and is the basis for this work. In this chapter the description of the framework and the contact models is enlarged. Due to changes in the contact detection, the expected time savings are accomplished and furthermore an in-depth analysis of new simulation results is conducted with an improved scenario.

The simulation framework is presented in Sect. 8.2. Details of the different contact models as well as a comprehensive comparison of their capabilities and applicability are given in Sect. 8.3. Heterogeneous contact modeling is the main idea of this work and is presented in Sect. 8.4. Alongside the explanation of the approach, a description of the virtual test setup used for evaluation is given. Simulation results and their discussion are shown in Sect. 8.5.

8.2 The DLR Rover Simulation Toolkit (RST) for Modelica

Modelica is a multi-physics, object oriented modeling language [3]. Base objects in multiple physical domains are defined by equations and interfaces, which more detailed components such as specific motors or mechanical parts can inherit from. This enables the modularity that is essential for the core part of this work while the numerous types of base objects enable modeling and subsequent simulation of complex multi-physics systems.

The RST is a Modelica library covering all relevant physical subsystems of a planetary rover, such as drivetrains, sensors and electrical systems. As shown in Fig. 8.1, other custom or commercial libraries like the DLR Visualization Library [4] are used together with RST components.

As the RST is especially focused on the locomotion and in particular on the wheel-ground contact, a dedicated Contact Dynamics Library (CDL) is used. This custom in-house library contains contact models with very different capabilities and applications which are described in more detail in Sect. 8.3. The contact models have a common interface (cf. Fig. 8.1), i.e. they require the position, velocities and orientation from the multi-body system (MBS) and respond with resulting forces and

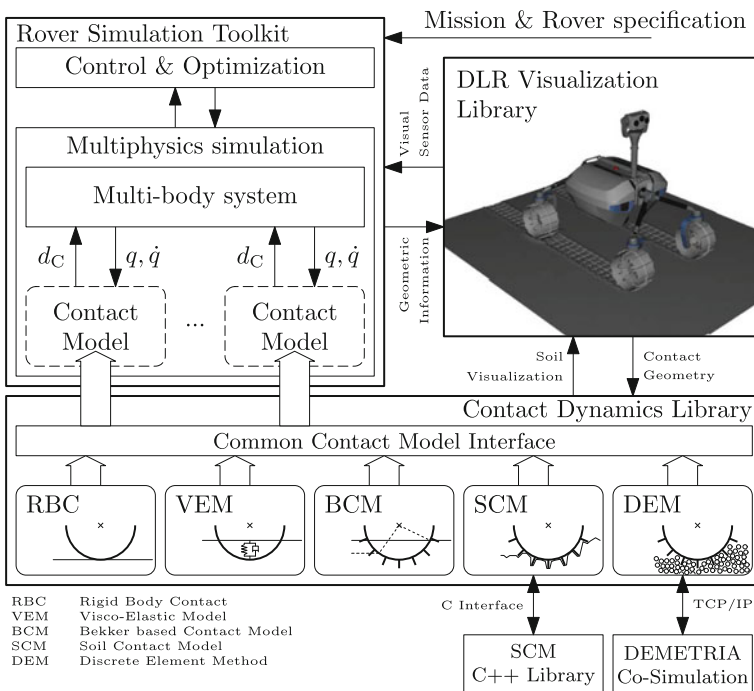


Fig. 8.1 Structure chart of the overall simulation framework, showing the interaction between its different parts

torques. The use of the CDL transfers its variability in contact modeling for different applications to the RST. These applications comprise but are not limited to high level mission demonstration, optimization of kinematic aspects, phase 0/A studies (cf. European Cooperation for Space Standardization) for planetary exploration missions, simulation-based forensic engineering, controller design and Hardware-/Software-in-the-loop (SIL/HIL) simulations.

8.3 Wheel-Ground Contact Models

In this section the wheel-ground contact models included in the Contact Dynamics Library (CDL) are presented and their main features, applications as well as advances compared to the state of the art are shown. In the end of the section a systematic comparison of the models with respect to typical simulation tasks in planetary exploration is given.

8.3.1 *Rigid Body Contact—RBC*

The simplest simulations of multi-body systems in conjunction with contact dynamics are typically based on rigid bodies only. While neglecting many effects of real world objects, the results are still sufficiently accurate for many applications focusing on the body movement. The big advantage of this approach is the short simulation time required: with modern desktop computers even large and complex contact scenarios can be simulated in real-time.

8.3.2 *Methods*

A rigid body is defined as an idealized, perfectly non-deformable object, independent of the external forces acting upon it. General constrained connections, joints and contacts always maintain their imposed constraints and hard impacts between objects cause instantaneous object speed changes to avoid any penetration. In reality the contact between two hard objects deforms both objects, even if only slightly, for a very short time before they flex back to their original form, separating the two objects. Since this effect happens within such a very short period of time, one might only be interested in the result of this contact, namely the change to the objects movement, due to the contact.

For a rigid body simulation only the impulse of a contact, is calculated and then reapplied to the objects, thereby instantaneously changing their velocity. For further details refer to [5–7]. This principle may be used to directly calculate the speed of the

two objects, right after a collision happened between them, according to the *Impulse based collision law*, modeling the dissipation by the *coefficient of restitution*.

The *coefficient of restitution* $\varepsilon \in [0, 1]$ depends on the material of the two objects involved. A value of 0 is called a perfect inelastic collision, a value of 1 corresponds to the perfect elastic collision.

Persistent contacts are modeled, similar to joints, as constraints on the possible relative movement. However, they only limit the movement in one direction. This is typically formulated as a *Linear Complementarity Problem* [8]:

$$\mu = A\lambda + b \quad (8.1)$$

$$\mu_i \geq 0 \quad \forall i \quad (8.2)$$

$$\lambda_i \geq 0 \quad \forall i \quad (8.3)$$

$$0 = \sum_i \mu_i \lambda_i \quad (8.4)$$

$$\mu, \lambda, b \in \mathbb{R}^n; A \in \mathbb{R}^{n \times n}; i \in [1, n]_{\mathbb{N}}$$

Applied to the contact modeling this means: either the relative force is larger than zero and the relative acceleration is zero, that is when the object surfaces are in touch, or the relative acceleration is larger than zero and the relative force is zero, when the objects are separating [9].

While dynamic friction may easily be incorporated into this approach as a force acting against the relative tangential motion of the objects, static friction is more challenging. The most common solution to this, and the one implemented here, is an approximation of the friction cone by a symmetric polyhedron [7].

8.3.3 Application

Rigid body models are one of the simulation techniques used most often in the computer animation and gaming industry. However, for scientific simulations its accuracy imposes certain limitations that need to be considered.

In the context of wheel ground contact simulations this technique is not applicable to the contact with soft soil, yet it is well suited for the contact between rigid wheels and a hard surfaces. In a heterogeneous simulation environment, the high computational efficiency of this model can therefore be used to quickly simulate the contact of wheels with stones in the ground.

8.3.4 Visco-Elastic Model—VEM

The foundation of the Visco-elastic model is that at the single point of collision a virtual spring-damper—also named KELVIN-element—is introduced. By attaching the spring-damper to the contact points of the overlapping bodies, deformation imposed forces are modeled. The difference between the non-penetrating rigid-body model and this penetration-based model is depicted in Fig. 8.2.

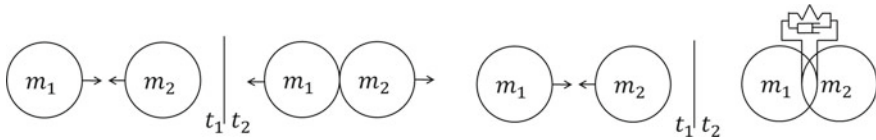


Fig. 8.2 Rigid-Body model and penetration model for Wheel-ground contact simulations—*Left side* impulse transfer, *right side* virtual spring damper elements

8.3.5 State of the art

In many non-scientific applications this approach is used with spring-damper parameters that are not representative for the objects involved but rather chosen in a way that the result gives a qualitative agreement to reality while maintaining numerical stability at larger time steps. Obviously this scaling approach is not suitable for a scientific simulation [9]. Besides a physical parameterization, more complex and realistic simulations may also incorporate the penetration volume and non-linear spring-damper characteristics (cf. [10]).

For the accurate modeling of real objects, sufficiently high spring and damper constants are required. During the simulation of several stiff bodies colliding, a very small integration time-step is required for the numerical stability of the resulting stiff system. Even for unconditionally stable integration schemes the penetration change during one step can lead to extremely high separating forces during the next step, causing unrealistic behavior if the time-steps are too large [7].

8.3.6 Methods

In the Contact Dynamics Library we use a spring-damper element based on penetration depth which can either be configured as a linear spring and damper or non-linearly based on Bekker's parameters (cf. Sect. 8.3.8) at critical damping in conjunction with LEHR's damping fraction. For the tangential contact force a COULOMB friction model

with optional static friction based on a non-linear function of the relative movement speed is implemented. This approach avoids the need for the common regularization using further KELVIN elements. The contact detection is based on the DLR Visualization Library and uses the same contact detection technique as shown in Sect. 8.3.8.

8.3.7 Application

Visco-Elastic models are popular where the simulated scene either isn't too complex and simulation speed is crucial or wherever the detailed evolution of contact forces is a required result and simulation time does not matter. Sufficient computational speed is achieved as long as the configuration is not too stiff as described in Sect. 8.3.4. The results are typically more accurate than those of the rigid-body simulation, especially if at least one of the involved bodies is comparatively soft. If realistic stiffness values are used in conjunction with non-linear spring-damper elements, high accuracy can be achieved by sacrificing computational efficiency [11]. The model is neither based on terramechanic considerations nor on impulse exchange, hence it is not the first choice for wheel-soil interaction focused simulations. However, with the correct parameterization this method can be used for some applications e.g. mission scenario demonstration to simulate both the hard contacts of wheels with rocks and for soft soil contacts maintaining a certain limited sinkage. For such applications the accuracy may be sufficient and its computational efficiency makes it a reasonable choice.

8.3.8 Bekker Based Contact Model—BCM

The main idea of the Bekker based Contact Model (BCM) is to provide the fundamental terramechanical effects on soft, sandy soils with low computational effort. The contact forces and torques are computed based on a single wheel reference point and a plane describing the contact situation to the actual surface. The wheel is solely described by a set of geometrical parameters, i.e. no point cloud or surface model is used. By applying the well known Bekker–Wong theory [15], BCM can be used in scenarios where specialized terramechanical effects like rutting or multipass can be neglected but effects like the sinkage behavior, slip or the maximum traction force cannot.

8.3.9 State of the art

Bekker's base model is best known for its role in the design of the Lunar Roving Vehicle [12], but has found popularity in the development of planetary exploration rovers. A similar implementation, a part of the ARTEMIS (Adams-based Rover

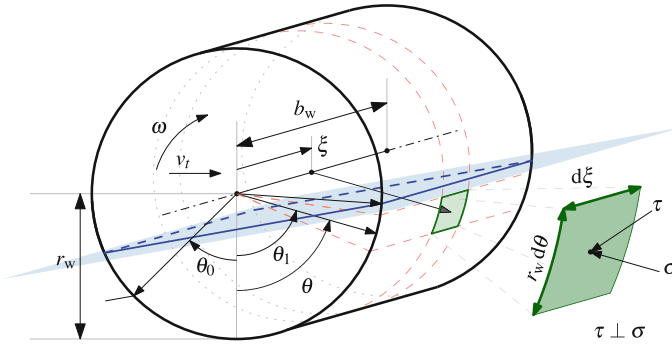


Fig. 8.3 Concept of BCM with an infinitesimal contact patch in green, the estimated soil plane in blue, the forward θ_1 and backward θ_0 contact angle as well as the control variables $\theta \in [\theta_0, \theta_1]$ and $\xi \in [0, b_w]$

Terramechanics and Mobility Interaction Simulator), was presented by Trease and the group of Iagnemma [13, 14]. The shown implementation was used to analyze the MER (Mars Exploration Rover) mobility. Like BCM these models are mainly based on the semi-empirical relations developed by Bekker in the 1950s with modification by [15]. Since Bekker's theory only describes the relation between sinkage and normal pressure it is commonly combined with Janosi and Hanamoto's extensions to Mohr–Coulomb failure criterion which is also described in [15].

8.3.10 Methods

In BCM the reaction forces and torques are calculated by evaluating the resulting normal and shear stress for single contact patches on the wheel surface in relation to the simplified soil plane (Fig. 8.3). Additionally, force reactions on the grouser faces and tips as well as the wheel sides are considered.

The soil is described by the Bekker parameters n , k_c , k_ϕ , the angle of internal friction ϕ , the macroscopic cohesion c , its bulk density ρ as well as three BCM specific parameters: V_j used in the Janosi–Hanamoto implementation of the shear stress and η , V_θ describing the contact geometry and its velocity dependency. The wheel is described by the radius r_w , width b_w , the grouser number n_g and height h_g .

BCM assumes an exponential reduction of the backward contact angle θ_0 with a maximum reduction of $\|\theta_0\| = \eta \|\theta_1\|$ (see Fig. 8.3). The assumed contact area spanned by $[\theta_0, \theta_1] \times [0, b_w]$ is divided into smaller patches of the size $A = r_w d\theta d\xi$. The total reaction force and torque is calculated by integrating over θ and ξ . The normal pressure σ acting on each contact patch is based on Bekker pressure sinkage relation [15] with b being equivalent to the wheel width b_w .

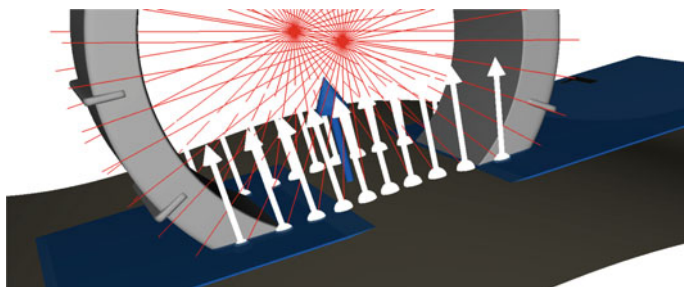


Fig. 8.4 Contact plane estimation in BCM implemented using the line surface contact detection provided by the DLR Visualization Library. With the search direction in *red*, the detected contact points and normals in *white* as well as the estimated soil plane in *blue*

$$\sigma = \left(\frac{k_c}{b} + k_\phi \right) z^n \quad (8.5)$$

The shear stress τ acting on each contact patch is calculated by reducing the maximum shear stress τ_{\max}

$$\tau = \tau_{\max} \operatorname{sgn} \left(\frac{dj}{dt} \right) \left(1 - e^{-\|dj/dt\|/V_j} \right) \quad (8.6)$$

This shear stress (τ)-shear velocity $\left(\frac{dj}{dt} \right)$ relationship is derived from the shear stress (τ)—shear displacement (j) relationship originally developed by Janosi–Hanamoto which proposes an exponential relationship [15]. The used maximum shear stress τ_{\max} is the result of the Mohr–Coulomb failure criterion

$$\tau_{\max} = c + \sigma \tan \phi. \quad (8.7)$$

Forces acting on grouser and the wheel sides are based on Rankine's passive earth pressure [16]

$$\sigma_p = \gamma_s z N_\phi + q N_\phi + 2c \sqrt{N_\phi}, \quad \text{with } N_\phi = \tan^2 \left(\frac{\pi}{4} + \frac{\phi}{2} \right), \quad \gamma_s = \rho g. \quad (8.8)$$

Figure 8.4 shows a visualization of the algorithm used to reduce the contact geometry to a plane. The algorithm is based on line surface contact detection provided by the DLR Visualization Library [4]. The contact plane is fitted into the detected contact points, contact normals, and the wheel normals corresponding with the contact points using a least squares optimization.

8.3.11 Application

The model is suitable wherever fast simulations covering the basic effects of terramechanics are required. Thus, it is most applicable for virtual prototyping in control, as used in the feasibility study for the DLR-SR robotic single wheel testbed [17].

8.3.12 Soil Contact Model—SCM

Up to this point the soil did not actually deform. However, one very important effect for the simulation of planetary rovers is the plastic deformation of soil caused by the wheels. In our simulation framework we use the SCM (Soil Contact Model) algorithm for the simulation of soft soil contact forces and plastic deformation of the soil. SCM is a in-house developed, highly specialized, three dimensional, novel extension of the well known Bekker–Wong method based on [1, 18].

8.3.13 State of the Art

Similar to BCM, SCM is based on Bekker’s theory and incorporates several extensions. In earlier implementations of SCM by Krenn [1, 18] the soil deformation was implemented similar to [19]. Another implementation of discretized Bekker model using soil deformation is shown in [13, 20]. An alternative approach to cover soil deformation in empirical soil models is shown in [21], by using locally spawned particles for the displaced soil volume.

8.3.14 Method

Analogously to the previously introduced models, SCM is based on surface contacts for both wheel and soil. In contrast to the single point contact models, which use the soil surface only for contact detection, SCM calculates reaction forces and the soil deformation by mapping the wheel nodes onto a discretized soil grid. Therefore, only the soil nodes in contact with the wheel grid are used. The normal and shear stress calculations are modular. In this context a normal model based on Bekker’s theory [15] is used. The shear model uses an implementation of Mohr–Coulomb failure criterion with extensions by Janosi–Hanamoto [15].

In order to cover plastic soil deformation and thus rutting, as visualized in Fig. 8.5, the distribution of the soil displaced by the wheel is based on flow. A novel approach based on theoretical soil mechanics is used to deposit the distributed soil onto the surrounding nodes. Afterwards an erosion algorithm is applied to all modified nodes

to ensure that the angle of repose is abided. Thus, landslides induced around the wheels can be covered by SCM's plastic soil deformation.

Using this approach, SCM enables to cover the main effects of terramechanics and soil deformation, namely bulldozing, rutting, multipass and slip sinkage in the environment of multi-body dynamics in an efficient way. Therein multipass and rutting are covered mainly geometrically and were recently enhanced. While the volume of disposed soil and its strength are influenced by a plasticity parameter, the soil parameters themselves remain unchanged.

Summarizing the main features of the enhanced SCM are:

- Surface contact with arbitrarily shaped objects,
- Z-Buffer contact detection for each node,
- contact pressure calculation for node in contact,
- modular normal and shear stress models (for example using Bekker–Wong theory and Mohr–Coulomb failure criterion),
- coverage of dynamic slip sinkage,
- plastic soil deformation covered by compaction and displacement,
- soil displacement and compaction by theoretical soil mechanics, flow field and erosion algorithm,
- simultaneous contact of multiple objects,
- and parallelization.

8.3.15 Application

labelsec:app3

SCM has been successfully used in the simulation of planetary rovers [18] and the evaluation of its control using multi-body dynamics [22]. A first verification of the model was carried out in [18] for models of pressure sinkage tests, as well as full-system scale tests. Further validation is currently performed using the DLR-RMC single wheel test facility.

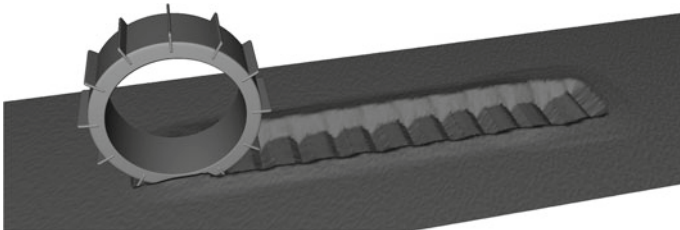


Fig. 8.5 SCM for Wheel-ground contact simulations including terrain deformation and rutting

8.3.16 *Discrete Element Method—DEMETRIA*

The most detailed models are based on particle methods, i.e. the Discrete Element Method (DEM). These methods allow to model regolith directly as granular matter without the need of empirical relations.

However, even for modern powerful computers, simulations using the real grain size is still not feasible.

8.3.17 *State of the Art*

The Discrete Element Method (DEM) was first announced by Cundall and Strack [23]. In the recent years the method was widely adapted, improved and used by many researchers ([24–26] a.o.). In order to model real soils, the most important adaptations are the coverage of the grain shape by either complex contact geometries, e.g. [26–28] or resistance torque laws, e.g. [24, 29, 30]. Additionally, the mapping of the particle parameters to real soils was only partially solved (e.g. [24]) or carried out by calibration [28, 31, 32]. Other fields of research try to improve the computational efficiency of the method [33, 34] or deal with the calculation of hard, non-penetrating, contacts [34–36] (see also Sect. 8.3.1).

In application for planetary rover wheels, DEM has been used in order to identify influences of wheel design parameters [25], the wheel performance in lunar environment [28] or to analyze NASA's Mars Exploration Rover (MER) wheel in towed configuration [26]. Another application for wheeled vehicles is shown in [37] focused on military offroad vehicles.

8.3.18 *Method*

DEM is based on inter-particle contact reaction and the solution of the equations of motion for every single particle in the simulation domain. Thereby the contact laws applied are crucial for the accuracy of the simulation results. In order to allow for precise but still efficient simulations, the DLR-SR particle dynamics framework “DEMETRIA” (Discrete Element Method Enabled Terramechanics Interaction framework), based on the particle simulator Pasimodo [38], is modeling the particle shape by additional rotation geometries. By using one of these two dimensional geometries per rotation axis, an equivalent 3D rotation primitive is formed [39]. Furthermore, the framework features a systematic particle scaling and a priori parameter estimation [40] as well as dynamic boundaries. These boundaries are moved together with the tool and minimize the active number of particles by loading and deleting particles on the fly [11]. For the macroscopic contact to the wheel, the same contact laws as for inter-particle interaction are applied. However, a different parameter set is used, since the material interface is different as well. In the end, the contact reactions

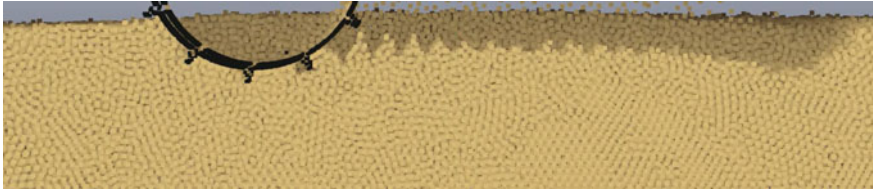


Fig. 8.6 Wheel-soil interaction in the particle-based wheel-soil Interaction model showing principle effects of soil deformation in terramechanics

are summed up and applied to the macroscopic wheel body. For the framework's main features and advantages the reader should refer to [40–42].

8.3.19 Applications

DEMETRIA was successfully applied to simulation of planetary rover wheels, exemplifying running surface optimization potential [43]. The basic effects like rutting, bulldozing and dynamic sinkage, of a certain wheel surface geometry are exemplified in Fig. 8.6. Additionally, it was applied to InSight's [44] subsurface locomotion system—a self impelling nail nicknamed “the Mole”: The HP³-Mole [45] was simulated using co-simulation of particle-based soil and the MBS mechanism model [42] and influences of the outer shape on the performance were shown [46]. This co-simulation is based on TCP/IP connection between the simulators and could be used with the RST in the same way. The particle-based soil models have been verified and validated using several kinds of material tests, usually used for characterization of soils [11, 47]. In addition to that the HP³-Mole's co-simulation results are validated against deep penetration tests with an error in predicted penetration depth of less than 16% [11]. The DEM wheel models have been checked for their qualitative behaviour in worst-case soil conditions and are currently being validated using the DLR-RMC single wheel test facility.

However, due to the high demand on computation time and power, DEM is not suitable for the simulations of long trajectories at full vehicle level. Thus, these models are mainly used in order to investigate and understand the low-level effects of the interaction and thereby to enrich more efficient models. Hence the particle-based models will not be used for further investigations in this article, but are considered for future heterogeneous wheel-ground contact studies.

Table 8.1 Comparison of used models of the Contact Dynamics Library

	Rigid body	Penetration based			
		VEM	BCM	SCM	DEM
Wheel stone contact	+ +	+	n/a	n/a	--
Wheel soft soil contact	n/a	--	--	+	+ +
Computational efficiency	+ +	+ +	+	--	--
Considered effects	--	--	+	+	+ +
Soil scalability	n/a	+ +	+ +	--	--
Multiple contacts	+ +	+ +	+ +	+ +	+
Soil deformation	n/a	n/a	n/a	+	+ +
Wheel description	S-mesh ^a	r_w, b_w	r_w, b_w, n_{Gr}, h_{Gr}	P-cloud ^b	S-mesh ^a
2nd body description	S-mesh ^a	S-mesh ^a	S-mesh ^a	E-map ^c	PFV ^d

^aSurface mesh^bPoint cloud^cElevation map^dParticle filled volume

8.3.20 Systematization of the Contact Models

One main advantage of the Contact Dynamics Library (models see Sects. 8.3.1–8.3.16) is that it enables easy exchange of the contact models. In order to determine which model to use for which application, a high-level overview of the models including a comparison of their characteristics is given in Table 8.1. It is pointed out that this table can only give a very general idea whereas for details the sections above need to be consulted. In typical scenarios in planetary exploration like the contact with deformable sandy soils or hard stones, each model features a certain level of detail for the application. As BCM and SCM are tailored to cover the soft soil contact only, they are not suitable for the application in wheel stone contacts. VEM and DEM are capable of covering this problem class using a different set of parameters—in both cases stones are bodies with a finite stiffness. However, rigid body approaches are most suitable for wheel stone contact, as the real deformation is negligible.

In terms of wheel sand contact, the number of considered effects is proportional with the complexity of the model. Thus DEM features the highest level of detail, due to the relocation based soil deformation. Scalability mainly depends on the discretization and dimension of the models. Thus VEM and BCM being one/two dimensional models scale best, whereas SCM and DEM being 2.5 and 3D methods slow down drastically with bigger domain sizes. Further effects add to the worse scaling behavior.

The row ‘multiple contacts’ indicates the scalability for multiple contact objects. All models but DEM scale linearly with the number of these objects. Hence, DEM is less suitable for full vehicle simulations than lower-tier methods, because of the excessive amount of computation time needed. Soil deformation is covered by the farther detailed terramechanics approaches only. Therein, SCM covers the soil defor-

mation by semi-empirical approaches, whereas DEM directly covers the deformation by grain relocation.

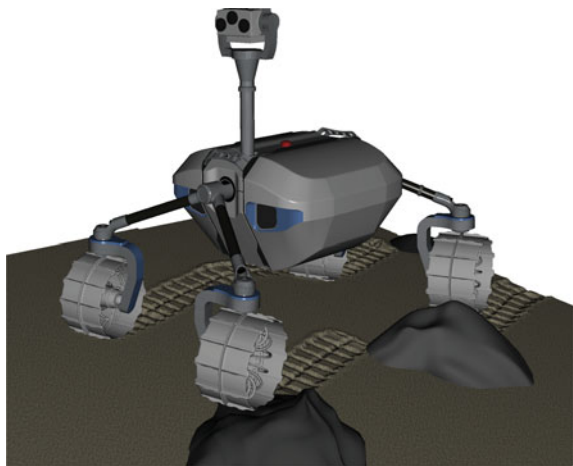
In SCM and DEM the wheel description is based on surface meshes described by vertices (nodes) and faces (elements) whereas in the other models a parametric description is used. Anyway, the surface meshes of the wheels are generated using the same parametric description, with certain limitations in terms of grouser geometry etc. for the purely parametric descriptions. The soil or general contact partner is described by surface meshes for rigid body, VEM and BCM as well. Only SCM, using equidistant and structured elevation maps and DEM using a particle filled volume described by position and orientation of each individual, are using different discretization approaches.

8.4 Heterogeneous Wheel-Ground Contact

The framework of the Rover Simulation Toolkit together with the Contact Dynamics Library enables using different contact models within one system model. This feature is used e.g. for the simulation of a rover traveling over sandy terrain with additional rocks embedded in the sand (cf. Fig. 8.7). For that scenario very heterogeneous contact models like SCM for the sand-contact and rigid body or the visco-elastic model for the rock-contact are used.

Besides this application which is motivated by different contact properties, a heterogeneous simulation may also be used to get a good trade-off between fast simulation and high level of detail. An example for such an approach is explained in the following and simulation results are shown in Sect. 8.5.

Fig. 8.7 DLR-RMC Lightweight Rover Unit (LRU) [48, 49] in rough terrain, wheel-sand contact modeled with SCM, wheel-rock contact modeled with VEM



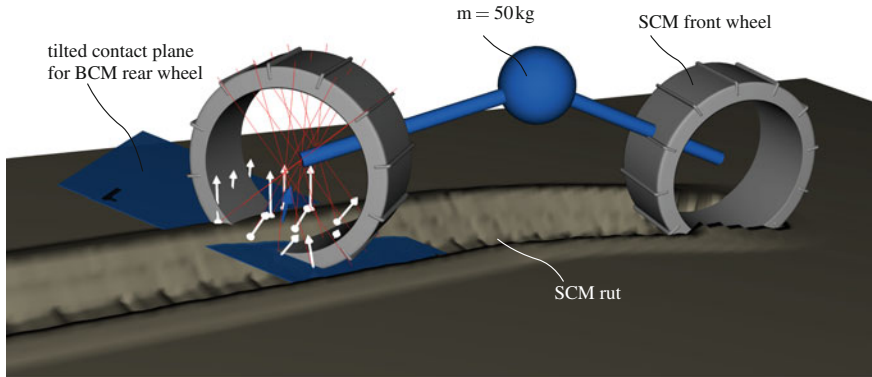


Fig. 8.8 The test setup with BCM's contact plane detection in the SCM rut

8.4.1 Approach

As outlined in Sect. 8.3 and especially in the comparison in Table 8.1, different contact models have distinct capabilities and profoundly different computation times. Knowing these specifications, coupling fast and slow models can drastically improve simulation times compared to homogeneous higher-tier models with acceptable influence on the result accuracy. To achieve that, each wheel's contact model needs to cover the major individual effects of its interaction with the ground. The leading wheels of a planetary rover are usually driving through untouched and potentially loose and uncompacted soils. Thus, their model needs to not only cover the current sinkage and reaction forces, but also the soil displacement causing additional resistance due to bulldozing, as well as the generation of ruts. These ruts can lower the trailing wheel's driving resistance and at the same time apply higher lateral guidance forces. In order to cover the rutting, SCM (cf. Sect. 8.3.12) is used for the rover's leading wheels, whereas the trailing wheels are modeled one tier lower as single point Bekker (cf. Sect. 8.3.8) in order to study the approach's reasonableness. The other models presented in Sect. 8.3 will not be used in this feasibility study, but may be considered for following investigations in the field of heterogeneous wheel ground contact. Figure 8.8 shows the main effect of the proposed heterogeneous contact which is to use SCM's deformed soil for the contact detection of the trailing wheels instead of the undeformed terrain. Thereby it will be shown that the force characteristics caused by the rut of the leading wheel correspond closely to a simulation with SCM for all wheels.

The influence of the used soil simulant's compressibility on the soil parameters for the trailing wheels is neglected in this first study. Moreover, the same soil parameters are used for both SCM and BCM where applicable.

Table 8.2 Parameters of the multi-body system, the wheel, the soil and the scenario in general

Parameter	Value
Wheel base	0.6 m
Location of point mass	10 cm above wheel axes (cf. Fig. 8.8)
Mass of point mass	50 kg
Free DoFs	All but the rotation about longitudinal rover axis
Rotational velocity of rear wheel steering	$[-0.8 \frac{\text{rad}}{\text{s}}, 0.8 \frac{\text{rad}}{\text{s}}]$
Rotational velocity of the wheels	$1 \frac{\text{rad}}{\text{s}}$
Wheel diameter d_w	250 mm
Wheel width b_w	125 mm
Grouser height h_{Gr}	10 mm
Number of grousers n_{Gr}	12
Mass of one wheel	2 kg
Soil density ρ	$1896 \frac{\text{kg}}{\text{m}^3}$
Angle of internal friction ϕ	36.7°
Cohesion c	66.5 pa
Bekker parameter ¹ k_c	$-2.86 \times 10^6 (\text{N/m})^{n+1}$
Bekker parameter ¹ k_ϕ	$2.47 \times 10^8 (\text{N/m})^{n+2}$
Bekker parameter ¹ n	2.49
Size of the soil plane	3 m × 1 m with 10 mm resolution (for SCM)

¹The used Bekker parameters are the result of the fitting method described in [50] and do not necessarily directly correspond with physical soil properties

Table 8.3 Simulation variables used for assessment and their corresponding objectives

Variable	Objective
CPU time factor k_{CPU}	Compare computation time
Tractive force F_x	Evaluate traction and its distribution on the front/rear wheel
Lateral force F_y	Evaluate guidance effect of the rut
z-position z	Evaluate the influence of the rut on the sinkage
Trajectory in the x - y -plane	Visualize impact of all above on the movement of the wheels

8.4.2 Virtual Test Setup

In this section, the test setup used for comparing the results of the different approaches in Sect. 8.5 is presented and choices and assumptions that were made are explained. Detailed parameters of the setup are given in Table 8.2 and the quantities that are used for evaluation are presented in Table 8.3.

The multi-body system For this study of heterogeneous wheel-ground contact modeling of a rover, we are mainly interested in effects of the rear wheels driving through ruts that were created by the front wheels. Using symmetry in conjunction with suit-

able boundary conditions we may simplify a four wheeled rover to only one front and one rear wheel which are connected by a rigid link and have a point mass located in between them (cf. blue body in Fig. 8.8). This assembly is able to move freely in all but the rotational degree of freedom about its longitudinal axis. Additionally, both wheels can be actuated, i.e. rotated about their local wheel axis and the rear wheel can be steered, i.e. rotated about its local z -Axis. The parameters can be found in Table 8.2.

The soil is a soil simulant for Martian regolith the so-called *MSS-D*. Its parameters were characterized using the DLR-RMC Bevameter in conjunction with the corresponding identification approach [50]. The parameter set is filed with the name RMCS-2. *MSS-D* was developed to simulate Martian regolith in terrestrial tests. Therefore, it mainly consists of fines and quartz sand. In addition to these soil parameters, both SCM and BCM feature a set of supplementary parameters which are chosen using physical and empirical assumptions. Additionally, the parameter choice made ensures comparability of the homogeneous simulations behavior. From a geometric point of view, a $3\text{ m} \times 1\text{ m}$ plane with a mesh resolution of $1\text{ cm} \times 1\text{ cm}$ is used for SCM and another one with a reduced mesh resolution of $3\text{ cm} \times 3\text{ cm}$ for BCM. The latter choice has an effect for the contact plane detection in the rut of a SCM wheel, only. The different resolutions were found to ensure a good compromise between computational effort and result quality. It is pointed out that the lower soil resolution for BCM is only possible because—in contrast to SCM—the geometry is solely used for calculating the current contact plane (see Sect. 8.3.8).

The wheels have a cylindrical shape with twelve grousers and beyond that a smooth surface; the parameters are summarized in Table 8.2.

The scenario In order to investigate the effects of ruts of the leading wheel, a scenario wherein the trailing wheel enters, escapes and crosses the trajectory of the leading wheel is chosen. The traces of both wheels as well as the setup itself and the dimension in x -direction can be seen in Fig. 8.9. Therefore, the two wheels start aligned and travel with the same constant rotational velocity. Shortly after entering the rut of the leading wheel, the trailing wheel is steered with a constant rotational velocity for a short distance and thereby escapes the rut. Subsequently, the trailing wheel is steered back again, with a constant rotational velocity such that it crosses the trace of the

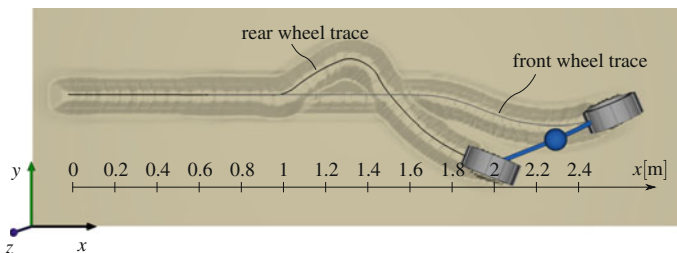


Fig. 8.9 Traces of front and rear wheel

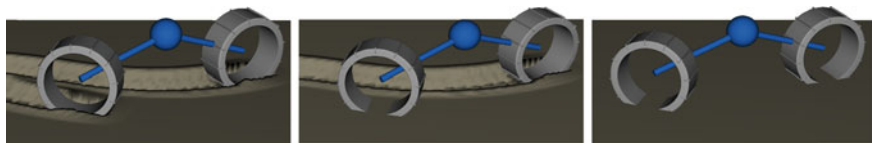


Fig. 8.10 Virtual test scenario to evaluate the tractive performance of different rovers: SCM-SCM (*left*), Heterogeneous SCM-BCM (*middle*), BCM-BCM (*right*),

leading wheel. The timing of all steering commands in the simulation is based on position thresholds in the global x -direction of the rear wheel.

The software For all shown simulations, Dymola 2016 RC-2 with a development version of the DLR Visualization library [4] and the Contact Dynamics Library is used. Furthermore, an explicit 4th order Runge–Kutta fixed step solver (*rkfix4* in Dymola) with a time step size of $\Delta t = 1$ ms is used. As the integration scheme does neither feature A-stability [51] nor step size control, the choice of the time step is constrained by the maximum eigen frequency of the system. A linearization of the contact stiffness according to Bekker’s equations (cf. Sect. 8.3.8) yields an eigen frequency of 76.5 Hz and thus a maximum time step size of 26 ms. This yields a safety factor of >10 for $\Delta t = 1$ ms. The choice of the solver itself is based on experience for a good trade-off between result quality and computational effort.

In order to check the consistency and the applicability as well as the potential speed up of the heterogeneous wheel ground contact modeling, a homogeneous simulation for each contact model is performed first. Figure 8.10 shows the single-tiered homogeneous, as well as the heterogeneous setups. For this evaluation the quantities and their respective evaluation objectives are listed in Table 8.3.

Since the objective of this work is to compare the coverage of basic terramechanics effects, the forces are normalized with respect to the maximum value of the longitudinal front wheel force of the homogeneous SCM simulation (for better readability of the plots, the force peaks at the start of the wheel rotation is not considered for this maximum value). These normalized values are noted as $(\cdot)_0$ in the following passages. The results of the described tests are shown in Sect. 8.5.

8.5 Results

In this section we discuss the impact of the proposed heterogeneous contact simulations (cf. Sect. 8.4). Therefore, we compare the accuracy of the simulation results as well as the demand in computation time of the homogeneous BCM and the heterogeneous simulation to the reference homogeneous high-tier SCM simulation. The values that are used for comparison are summarized in Table 8.3. These values are common quantities for tractive performance tests in planetary rover locomotion. All

forces are plotted in a local wheel coordinate system where the z -direction is co-directional with the global z -direction. The x - and y -axes are rotated with the wheel steering angle such that the x -axis points in the wheel's longitudinal direction at all times. Note that due to this rotation the forces in x - and y -direction only sum up to zero all together (i.e. not separated in x - and y -direction) for a stationary movement. The high frequency noise observable in the SCM force is a result of the soil discretization. Due to the uniform distribution and high frequency the effects can be neglected in this context.

8.5.1 Detailed Description of Observed Effects in the Different Setups

The longitudinal and lateral forces as well as the z -position of all three setups are shown in Fig. 8.12, top. Thereby, the z -position denotes the position of the wheel center above the undeformed ground level. It is pointed out again that all four force plots are normalized with respect to the maximum longitudinal force of the front wheel (the peak in the beginning is not taken into account since it is not of particular interest here and would shrink the rest of the plots). The trajectories in the x - y -plane of the front (solid lines) and rear (dashed line) wheels are shown in Fig. 8.12, bottom. In the following paragraphs the plotted results are explained in detail whereas a shorter overall summary of these results is given in Sect. 8.5.2. For better readability, the following abbreviations are used:

SxS homogeneous SCM model—both wheels SCM
 BxB homogeneous BCM model—both wheels BCM
 SxB heterogeneous model—front wheel SCM, rear wheel BCM

Also, the whole scenario can be divided into five main sections which are labeled with the letters A-E in the following and in Fig. 8.12:

- A Start of the wheel rotation and stationary driving.
- B The rear wheel enters the rut of the front wheel and subsequently drives in it.
- C The rear wheel is steered to $\delta = 0.8$ rad and quits the rut.
- D The rear wheel is steered back to $\delta = -0.8$ rad and crosses the rut.
- E Stationary driving of both wheels, each in its own lane.

0–0.1 m, *acceleration of the rover (A)*: There is a positive peak in front and rear wheel x -forces when the wheel starts to rotate. This is due to the correlation of high slip velocity and high shear stress. After the first 10 cm the plotted quantities seem to have reached a stationary state in all three models. However, to reach that state, SxS and SxB show a shift of the traction force to the rear wheel. This originates from dynamic wheel loading, i.e. a higher normal force on the rear wheel due to the inertia resulting from the point mass. It can also be seen that SxB transfers even more traction force to the rear wheel. This effect occurs due to the BCM simulated rear wheel being able to develop a higher traction force compared to the SCM front wheel. The latter also

shows increased bulldozing and hence lifting forces. This lifting of SCM-modeled wheels can also be seen in the z -position plot.

0.1–0.35 m, *stationary driving (A)*: All three models show a stationary condition, as rear and front forces are both almost canceled out. Looking at the z -position it can be seen that the sinkage is higher for SCM modeled wheels than for the BCM ones. This effect occurs due to differences in elastic and plastic deformation of the soil in the two models. However, the effect is beyond the scope of this work and will be subject of future research.

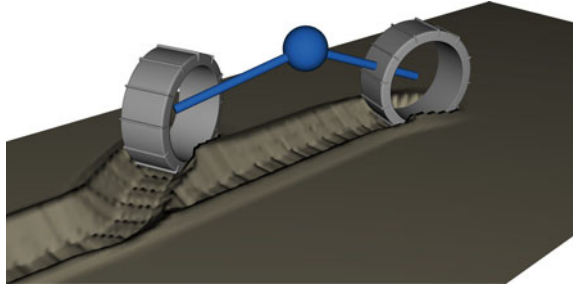
0.35–0.65 m, *rear wheel entering the rut (A-B)*: SxS and SxB both show a shift in the traction force distribution due to the rear wheel entering the rut of the front wheel. In SxS however, the rear wheel needs to traverse a bit of compacted soil first which leads to increased resistance at the rear wheel and a higher traction force at the front wheel accordingly. The BCM simulated rear wheel of SxB is able to perceive geometric changes only, which are very low for the region right before entering the rut. Hence, only SxB shows the pushing effect on the rear wheel when entering the rut. This effect can be observed in the Fig. 8.12 (top) as a shift of the tractive force to the rear wheel. By analysis of the z -position it can be seen that the rear wheel approaches the z -position of the SCM wheel by rolling down into the rut. BxB in contrast does not change from its stationary condition in any of the quantities, due to the non-existent rut of the front wheel.

0.65–0.9 m, *stationary driving in rut (B)*: The rear wheel drives stationary in the track of the front wheel. Besides the sinkage, all models deliver similar results.

0.9–1 m, *rear wheel steering (C)*: Constant steering angular velocity ($\dot{\delta} = 0.8$ rad) until an angle of $\delta = 45^\circ$ is reached. All models show the changing force distributions in the x - y -plane, i.e. the traction force of the front wheel increases. This is compensated by a higher resistance of the rear wheel both in longitudinal (x) and lateral (y) direction. Note that in the first instance of steering the wheels are still almost aligned and thus no lateral force due to a steered rear wheel is exerted. Rather a small lateral force in the opposite direction of the pulling is created by the steering itself. This effect is covered by all model combinations, although SxS shows a higher magnitude of the effect. In contrast to the other models the BCM rear wheel in SxB starts to climb out of the rut, whereas the SxS rear wheel even digs in a little deeper when the steering velocity is applied. SxS's higher lateral force (cf. $(F_y)_0$ in Fig. 8.12, top) and higher sinkage together result in a first small difference in the x - y -trajectory, i.e. the SCM rear wheel shows a delay in its y -position.

1–1.1 m, *rear wheel quitting the rut (C)*: In both SxS and SxB the resistance force at the rear wheel increases because it needs to drive out of the rut. In contrast, BxB's forces remain at the level that is given by a pure geometric correlation. To be more precise, the force distribution of longitudinal and lateral forces of front and rear wheel are given solely by the steering angles of front and rear wheel. The BCM rear wheel of SxB continues climbing out of the rut and reaches an even higher z -position than for BxB which is caused by the piled soil around the SCM front wheel's rut. However, since the SxB rear wheel needs to climb up the complete sidewall of the rut, it shows a stronger guidance in y -position compared to both homogeneous

Fig. 8.11 The homogeneous SCM configuration at rear wheel position 1.2 m where it starts steering back



simulations. This effect can be observed in the x - y -trajectory in Fig. 8.12 (bottom) where the rear wheel of SxB starts to diverge in the y -position from the other two.

1.1–1.2 m, *constant conditions (C)*: For this position range the forces of SxS and SxB converge towards the BxB results. However, the period of constant steering angle is not quite long enough for the former two to reach the BxB results.

1.2–1.4 m, *rear wheel steering (D)*: The rear wheel is steered back to $\delta = -45^\circ$ with $\dot{\delta} = -0.8 \frac{\text{rad}}{\text{s}}$ which leads to a corresponding change in the force distributions for all models. The z -position plot shows that the SxS rear wheel starts digging in as soon as the steering is applied at 1.2 m (cf. description of 0.9–1 m). As shown in the x - y -trajectory plot at the corresponding position this is not due to the rut of the first wheel since the rut is not reached yet, see Fig. 8.11. Thus, this lowering in z -position is pure sinkage due to the added resistance by steering. At 1.3 m the rear wheel is aligned with the front wheel again, which can be seen in the tractive forces of front and rear wheel being close to zero for a short distance. The SxS rear wheel even continues to sink in for about 4 cm after the alignment of the wheels was reached at 1.3 m. This considerable difference in sinkage leads to major differences in the forces in [1.3 m, 1.4 m], too. It can be seen that the SxS rear wheel is experiencing an increased resistance in both lateral and longitudinal direction. In this part the SxB result is even farther apart from the reference SxS than the simpler BxB. That can be seen looking at the rear wheel of SxB, which shows even lower resistance forces than BxB. This difference can be explained using the z -position plot: The rear wheel of SxB slides down the sidewall into the rut of the SCM front wheel and pushes the front wheel.

1.4–1.6 m, *rear wheel crossing the rut (D)*: In the SxB case the rear wheel needs to climb up the steep sidewall of the rut again. Thereby, it exerts an increased guidance force which results in a significantly lower gradient of the x - y -trajectory in Fig. 8.12 (bottom). The steep slope additionally causes the longitudinal resistance force being higher for this case. All together this leads to a significantly increased traction force on the front wheel. In contrast to that behavior, SxS starts to overcome the rather high force shift of its rear wheel that was explained above. As soon as the front wheel rut is reached, the rear wheel is driving on pre-compressed soil which also supports to lower its resistance force immediately.

Table 8.4 Factors for computation time

Model	Computation time factor
BCM	1
SCM	2.3
Heterogeneous	1.5

>1.6 m, *stationary driving (E)*: All three setups reach a similar stationary force distribution. Due to the differences described above, the trajectories continue to deteriorate. I.e. the whole rover steering angle of SxB is shifted compared to the homogeneous simulations. This is mainly an effect of the higher guidance force of the front wheel's rut on the rear wheel.

The intention of the heterogeneous contact modeling is to reduce computation time while covering as many details in the simulation result as possible. Hence, additionally to the result accuracy, the computation time is compared in Table 8.4. Factors for the computation time are introduced which represent the computation time of each setup normalized with the computation time of the BxB case, $k_{\text{CPU}}^{\text{modelX}} = t_{\text{CPU}}^{\text{modelX}} / t_{\text{CPU}}^{\text{BCM}}$. It can be seen that SxS takes more than twice the time compared to BxB whereas the SxB time is slightly better than the mean of SxS and BxB. Compared to the results in [2], we execute our contact search for the BCM wheels on a small patch of the surface only, which is deformed by the SCM front wheel. This leads to a major speedup of the heterogeneous model as was expected in [2]. Furthermore, due to a different scenario, modified contact models and a different solver the SCM is, compared to BCM, not as slow as it was in the previous work. Also consider that the simulations shown, where computed on a standard office PC. The simulation time would probably differ for other configurations e.g. due to SCM's recently added multi-threading capabilities.

8.5.2 Interpretation of the Results

The detailed explanation of all observable effects is given in Sect. 8.5.1, this short section is intended to briefly summarize the results and give an interpretation and implications for the different models/setup.

- For mission planning or similar applications where the position is the required result and variables only matter in their order of magnitude, a homogeneous BCM simulation provides adequate and sufficiently precise results. Moreover, the homogeneous BCM results for the wheel's trajectory are even closer to the reference SCM solution than the heterogeneous ones. This is caused due to the effects described after the next bullet point. Hence, the benefit of using farther detailed models for these applications should always be evaluated beforehand.
- BCM can neither model rutting of wheels nor changes in parameters for multi-pass simulation. If the effect of these ruts on the wheel forces is important, the

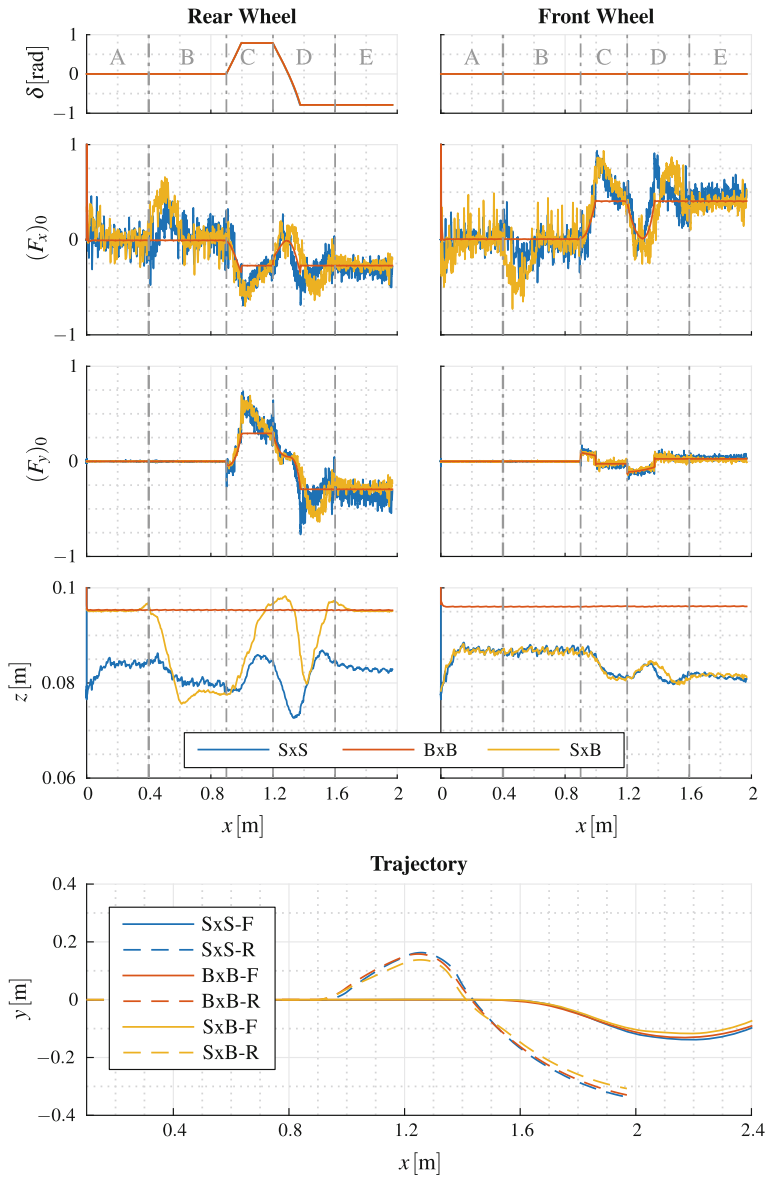


Fig. 8.12 Forces in longitudinal and lateral direction, z -position as well as the trajectory in x -plane of the front and rear wheels. (SxS: homogeneous SCM, BxB: homogeneous BCM, SxB: heterogeneous)

heterogeneous approach offers results that are very close to a homogeneous high-tier simulation while saving a considerable amount computation time. The saved time might be crucial in many applications in planetary exploration, e.g. due to tight schedules or for simulation based forensic-engineering [52].

- The main differences between the homogeneous high-tier SCM and the heterogeneous SCM/BCM forces result from the fact that BCM or—to be more precise—its contact detection is only able to detect and react on geometrical changes of the front wheel's rut. Hence, it needs to climb comparatively large sidewalls when crossing a rut compared to SCM digging through the sidewalls. Thus SCM does not need to lift the whole rover as much as BCM.
- Independent from the scope of this work, the sinkage of BCM and SCM was found to be different by approximately 25%. This will be subject to further investigation within the currently ongoing model validation campaign using our DLR-RMC single wheel test facilities.

To conclude this section, it should be mentioned again that none of the used models has undergone a in-depth validation yet, as this is one of our currently ongoing projects. However, in this work SCM is used as a reference for two reasons: First it is partially verified by previous analysis and second it is able to cover the most effects of the two models used. Since the scope of this section is to compare the qualitative capability of modeling certain effects of wheel-soil contact using homogeneous and heterogeneous approaches and study their qualitative effects on a rover, this approach is applicable and does not require in-depth validation a-priori.

8.6 Conclusion

In the article we presented the integration of wheel ground contact models with different level of detail in a unified simulation framework to allow for appropriate simulation of the various tasks in planetary exploration. Furthermore, this integration enables multi-tiered heterogeneous wheel ground contact modeling in a unified manner.

By usage of our single-point (BCM) and multi-point (SCM) Bekker-based contact models this approach was exemplified in order to achieve a speed up of the simulation compared to the homogeneous higher-tier SCM model. In the chosen planetary rover locomotion scenario, the computation time was decreased considerably while maintaining almost the same level of accuracy. Additionally, drawbacks and limitations of the approach were pointed out. Due to usage of SCM deformed surface patches of limited size in BCM's contact detection the speed up of the heterogeneous model is in the expected range in contrast to the results presented in [2].

As simulants, like MSS-D, which are mainly based on fines feature excessive compressibility, a next step will be the investigation of multi-pass effects in pre-compressed ruts in compressible simulants. Additionally, further validation of the single models as well as investigations will be performed. Moreover, in order to

allow a deeper insight in the potential speed up, rovers with increasing number of wheels, featuring leading wheel's SCM contact and lower-tiered contacts for the trailing wheels, will be compared in future work. It is expected that the benefit of the increased accuracy of the soil interaction models is decreasing with a higher number of the multi-passes.

Acknowledgments Parts of this work have been granted by the Helmholtz-Gemeinschaft Deutscher Forschungszentren e.V. under contract number HA-304 (Robotic Exploration of Extreme Environments—ROBEX)

References

1. Krenn R, Hirzinger G (2008) Simulation of rover locomotion on sandy terrain-modeling, verification and validation. In: 10th ESA workshop on advanced space technologies for robotics and automation—ASTRA, (2008) Noordwijk. Niederlande, ESA
2. Lichtenheldt R, Hellerer M, Barthelmes S, Buse F (2015) Heterogeneous, multi-tier wheel ground contact simulation for planetary exploration ISBN 978-84-944244-0-3
3. Elmqvist H, Mattsson SE, Otter M (1999) Modelica a language for physical system modeling, visualization and interaction. In: Proceedings of the 1999 IEEE international symposium on computer aided control system design (Cat. No.99TH8404)
4. Hellerer M, Bellmann T, Schlegel F (2014) The DLR Visualization Library Recent development and applications. In: Proceedings of the 10th international modelica conference—Lund, Sweden - Mar 10–12, 2014, pp 899–911
5. Erleben K (2004) Stable, robust, and versatile multibody dynamics animation. Phd thesis, University of Copenhagen, Denmark
6. Mirtich BV (1996) Impulse-based dynamic simulation of rigid body systems. Dissertation, University of California at Berkeley
7. Bender J, Erleben K, Trinkle JC (2014) Interactive simulation of rigid body dynamics in computer graphics. *Comput Gr Forum* 33(1):246–270
8. Cottle RW, Dantzig GB (1968) Complementary pivot theory of mathematical programming. *Linear Algebra Appl* 1(1):103–125
9. Boeing A, Bräunl T (2007) Evaluation of real-time physics simulation systems. In: Proceedings of the 5th international conference on computer graphics and interactive techniques in Australia and Southeast Asia, ACM pp 281–288
10. Marhefka DW, Orin DE (1996) Simulation of contact using a nonlinear damping model. *Proc IEEE Int Conf Robot Autom* 2(April):1662–1668
11. Lichtenheldt R, Schäfer B, Krömer O (2014) Hammering beneath the surface of mars— modeling and simulation of the impact-driven locomotion of the hp3-mole by coupling enhanced multi-body dynamics and discrete element method. In: Shaping the future by engineering: 58th Ilmenau scientific colloquium IWK, URN (Paper): <http://nbn-resolving.de/urn:nbn:de:gbv:ilm1-2014iwk-155:2> Technische Universität Ilmenau, 08–12 Sept 2014
12. Vivake Asnani, Damon Delap, Colin Creager (2009) The development of wheels for the Lunar Roving Vehicle. *J Terramech* 46(3):89–103
13. Trease B, Arvidson RE, Lindemann R, Bennett K, Feng Z, Iagnemma K, Senatore C, Van Dyke L (2011) Dynamic modeling and soil mechanics for path planning. In: Proceedings of the ASME 2011 international design engineering technical conference & computers and information in engineering conference IDETC/CIE, pp 1–11
14. Iagnemma K, Senatore C, Trease B (2011) Terramechanics modeling of Mars surface exploration rovers for simulation and parameter estimation. In: Proceedings of the IDETC/CIE 2011

- ASME international design engineering technical conferences & computers and information in engineering conference, pp 1–8
15. Wong JY (2008) *Theory of ground vehicles*, 4th edn. Wiley, New Jersey
 16. Terzaghi K, Peck RB, Mesri G (1996) *Soil mechanics in engineering and practice*. Wiley, New Jersey
 17. Fabian Buse (2015) *Masterarbeit Machbarkeitsstudie für einen roboterbasierten Radprüfstand zur Entwicklung von Mars Mondrovern*. Masterthesis, RWTH Aachen
 18. Krenn R, Hirzinger G (2009) SCM a soil contact model for multi-body system simulations. In: 11th European regional conference of the international society for terrain-vehicle systems–ISTVS 2009, Bremen, Germany
 19. Sumner RW, O'Brien JF, Hodgins JK (1999) Animating sand, mud, and snow. *Comput Gr Forum* 18:17–26
 20. Zhou F, Arvidson RE, Bennett K, Trease B, Lindemann R, Bellutta P, Iagnemma K, Senatore C (2014) Simulations of mars rover traverses. *J Field Robot* 31(1):141–160
 21. Holz D, Azimi A, Teichmann M, Kovacs J (2012) Mobility prediction of rovers on soft terrain: effects of wheel- and tool-induced terrain deformations. *Proceedings of the fifteenth international conference on climbing and walking robots and the support technologies for mobile machines CLAWAR 2012*:647–654
 22. Krenn R, Gibbesch A, Binet G, Bemporad A (2013) Model predictive traction and steering control of planetary rovers. In: 12th symposium on advanced space technologies in robotics and automation: ASTRA 2013
 23. Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. *Geotechnique* 29:47–65
 24. Obermayr M (2013) *Prediction of load data for construction equipment using the discrete element method*. PhD thesis, Universität Stuttgart
 25. Nakashima H, Oida A, Momozu M, Kawase Y, Kanamori H (2007) Parametric analysis of lugged wheel performance for a lunar microrover by means of dem. *J Terramech* 44(2): 153–162
 26. Knuth MA, Johnson JB, Hopkins MA, Sullivan RJ, Moore JM (2011) Discrete element modeling of a mars exploration rover wheel in granular material. *J Terramech*
 27. Das N (2007) *Modeling three-dimensional shape of sand grains using discrete element method*. PhD thesis, University of Florida
 28. Li W, Huang Y, Cui Y, Dong S, Wang J (2010) Trafficability analysis of lunar mare terrain by means of the discrete element method for wheeled rover locomotion. *J Terramech* 47(3):161–172
 29. Oda M, Iwashita K (2000) Study on couple stress and shear band development in granular media based on numerical simulation analyses. *Int J Eng Sci* 38:1713–1740
 30. Plassiard J-P, Belheine N, Donze F-V (2007) Calibration procedure for spherical discrete elements using a local moment law. Technical report, Universität Grenoble
 31. van der Linde J (2007) *Discrete element modeling of a vibratory subsoiler*. Master's thesis, University of Stellenbosch, Department of Mechanical and Mechatronic Engineering
 32. Asaf Z, Rubinstein D, Shmulevich I (2006) Evaluation of link-track performances using dem. *J Terramech* 43(2):141–161
 33. Harada T, Tanaka M, Koshizuka S, Kawaguchi Y (2007) Real-time coupling of fluids and rigid bodies. In *APCOM*, Kyoto
 34. Li A, Melanz D, Serban R, Negrut D (2014) A gpu-based preconditioned newton-krylov solver for flexible multibody dynamics. In: 3rd joint international conference on multibody system dynamics, IMSD 2014, Busan, Korea
 35. Negrut D, Tasora A, Anitescu M, Mazhar H (2011) Solving large multi-body dynamics problems on the GPU. In: *GPU Gems*, pp 269–280
 36. Kleinert J, Obermayr M, Balzer M (2013) Modeling of large scale granular systems using the discrete element method and the non-smooth contact dynamics method: a comparison. In: *ECCOMAS multibody dynamics* pp 1–4 July, University of Zagreb, Kroatien

37. Melanz D, Mazhar H, Negrut D (2014) Gauging military vehicle mobility through many-body dynamics simulation. In: The 3rd joint international conference on multibody system dynamics (IMSD 2014), Busan, Korea
38. Fleissner F (2012) Dokumentation, template-files und beispiele zum programmpaket "pasi-modo". Inpartik & Universität Stuttgart, Template files
39. Lichtenheldt R, Schäfer B (2013) Planetary rover locomotion on soft granular soils - efficient adaption of the rolling behaviour of nonspherical grains for discrete element simulations. In: 3rd international conference on particle-based methods, pp 807–818, ISBN 978-84-941531-8-1, Stuttgart
40. Lichtenheldt R (2015) A novel systematic method to estimate the contact parameters of particles in discrete element simulations of soil. In: 4th international conference on particle-based methods - particles 2015, pp 430-441, ISBN 978-84-944244-7-2, Barcelona
41. Lichtenheldt R, Schäfer B (2013) Planetary rover locomotion on soft granular Soils—efficient adaption of the rolling behaviour of nonspherical grains for discrete element simulations. In: 3rd international conference on particle-based methods, pp 807–818, ISBN 978-84-941531-8-1, Stuttgart, Germany
42. Lichtenheldt R, Schäfer B, Olaf K (2014) Hammering beneath the surface of mars—modeling and simulation of the impact-driven locomotion of the HP3-Mole by coupling enhanced multi-body dynamics and discrete element method. In: 58th Ilmenau scientific colloquium (IWK), Ilmenau, Germany, Technische Universität Ilmenau
43. Lichtenheldt R, Schäfer B (2013) Locomotion on soft granular soils: a discrete element based approach for simulations in planetary exploration. In: 12th symposium on advanced space technologies in robotics and automation: ASTRA 2013, Noordwijk, the Netherlands
44. Barnerdt WB et al. (2013) Insight: a discovery mission to explore the interior of mars. In: 44th lunar and planetary science conference, Texas, USA
45. Spohn T, Grott M, Smrekar S, Krause C, Hudson TL (2014) Measuring the martian heat flow using the heat flow and physical properties package (hp3). In: 45th lunar and planetary science conference
46. Lichtenheldt R (2015) Hammering beneath the surface of Mars—Analyse des Schlagzyklus und der äußeren Form des HP3-Mole mit Hilfe der Diskrete Elemente Methode. In: IFToMM D-A-CH 2015, Dortmund, Germany ISBN 978-3-940402-03-5
47. Lichtenheldt R, Schäfer B (2013) Locomotion on soft granular soils: a discrete element based approach for simulations in planetary exploration. In: 12th symposium on advanced space technologies in robotics and automation, ESA/ESTEC, Netherlands
48. Wedler A, Rebele B, Reill J, Suppa M, Hirschmüller H, Brand C, Schuster M, Vodermayr B, Gmeiner H, Maier A, Willberg B, Busmann K, Wappler F, Hellerer M, Lichtenheldt R (2015) LRU - lightweight rover unit. In Proceedings of the 13th symposium on advanced space technologies in robotics and automation (ASTRA)
49. Wedler A, Hellerer M, Rebele B, Gmeiner H, Vodermayr B, Bellmann T, Barthelmes S, Lange C, Witte L, Schmitz N, Knapmeyer M, Czeluschke A, Thomsen L, Waldmann C, Wilde M, Takei Y (2015) Robex components and methods for the planetary exploration demonstration mission (1). In: 13th symposium on advanced space 820 technologies in robotics and automation, ESA/ESTEC, Netherlands
50. Apfelbeck M, Kuß S, Rebele B, Schäfer B (2011) A systematic approach to reliably characterize soils based on bevameter testing. *J Terramech Elsevier* 48:360–371
51. Dahlquist G (1963) A special stability problem for linear multistep methods. *BIT* 3(1):27–43
52. Lichtenheldt R, Schäfer B, Krömer O, van Zoest T (2014) Hammering beneath the surface of Mars - forensic engineering of failures in the HP3-Mole by applying multi-body dynamics simulation. In: Proceedings of 3rd international conference on multibody system dynamics IMSD 2014, ISBN 978-89-950027-7-3, Busan, Korea

Chapter 9

Intervention-Autonomous Underwater Vehicle Multibody Models for Dynamic Manipulation Tasks

Roberto Conti, Riccardo Costanzi, Francesco Fanelli, Enrico Meli, Alessandro Ridolfi and Benedetto Allotta

Abstract Free floating autonomous underwater manipulation is still an open research topic; an important challenge is offered by dynamic manipulation, where the vehicle maintains relevant velocities during manipulation tasks. To develop new control architectures, a precise modelling of the mechanisms involved in the manipulation tasks is needed. The focus of this paper is the multibody modelling and the control of an Intervention-Autonomous Underwater Vehicle (I-AUV). An accurate model of the whole system has been developed, including vehicle-fluid interaction. A suitable 3D contact model has been developed for the contact between the gripper and the object to be manipulated. A control strategy for the whole I-AUV system is proposed, comprising a suitable grasp planning strategy. Finally, an evaluation of the I-AUV control system performances have been carried out.

Keywords Multibody models · Underwater manipulation · Autonomous underwater vehicles

9.1 Introduction

Nowadays Autonomous Underwater Vehicles (AUVs) are quite widespread. These vehicles can undoubtedly lead to substantial economic and technological benefits. In the technical evolution of the AUVs the following important topics are still characterized by many open problems: the dynamic performances and the control of the vehicle, the mobile tele-manipulation of a single vehicle (with relevant vehicle velocity) and the cooperation among vehicles. In this paper, the modelling and the control architecture of an AUV specifically thought for the underwater mobile manipulation, usually called I-AUV (Intervention-AUV), are described.

Currently, a considerable number of operations in sea-rescue, research and maintenance of oil rig appliances, got ahead using Unmanned Underwater Vehicles (UUVs),

R. Conti (✉) · R. Costanzi · F. Fanelli · E. Meli · A. Ridolfi · B. Allotta
Department of Industrial Engineering (DIEF), University of Florence (UNIFI),
Via di Santa Marta 3, 50139 Florence, Italy
e-mail: roberto.conti@unifi.it

need manipulation capacity to be concluded successfully [1–7]. In a such scenario, most of the intervention missions at high depths are faced up by remotely controlled vehicles equipped with one or more robotic arms (Intervention-ROVs), representing until today the standard technology in that field [8]. The ROVs for assistance, which can be teleoperated for long periods, are usually controlled with a master-slave approach [5, 6]. This kind of strategy has some limitations: the operator must be skilled with special type of training, underwater communication is often difficult and a significant delay in the control loop can be present. To overcome these limitations, many researchers are now focused on what appears as the AUV natural evolution, i.e. the autonomous underwater vehicles equipped with manipulator arms, the I-AUVs [4, 7–15].

One important contribution to the development of the state of the art of the I-AUV is due to TRIDENT, an European project lasting for 3 years and started in 2010 [4, 16]. The aim of TRIDENT was the development of new methodologies to complete manipulation assistances in non-structured underwater environments, through a cooperative team composed of an AUV equipped with a robotic arm at 7 degrees of freedoms (DOFs) and an ASC (Autonomous Surface Craft): the latter is an autonomous surface vehicle, whose aim is to replace, in the application near to the coast, the support ship with crew necessary to the running of the AUV. In March 2014 PANDORA project [17] has demonstrated free floating grasping and valve turning in tank. However in both cases the vehicle is in a hovering phase and not in “mobile navigation”. In fact, autonomous underwater robotic manipulation with free-floating base is far from reaching an industrial product. This is particularly true in the framework of dynamic manipulation, where relevant vehicle velocities are required (in contrast with hovering manipulation).

Concerning control strategies, the problem is still open as well. Vehicle-manipulator decoupled control strategies have been mostly studied until now, which independently control the AUV and the robotic arm [8, 15]; these strategies offer simpler hardware and software implementation and require less knowledge of the system parameters compared to arm-vehicle coupled control techniques [2, 8].

In this paper, a detailed 3D multibody model of the I-AUV system (vehicle, arm, gripper, object to be manipulated and fluid interaction [1, 18]) has been developed to test the proposed control strategy. Moreover, a suitable 3D contact model has been developed for the contact between the gripper and the object to be manipulated. For what concerns the control technique, a decoupled vehicle-manipulator strategy has been employed [2, 8]. This kind of techniques offers simpler hardware implementation and is more robust against the knowledge of the system parameters with respect to arm-vehicle coupled strategies. In addition, exploiting the hand kinematics, the control of the gripper has been further decoupled from the arm control: this way, the performances of the I-AUV are improved while maintaining higher vehicle velocities. Furthermore, a grasp planning algorithm, based on optical cameras [19], is proposed.

The models and the control architecture have been validated simulating a suitable test case using the software Matlab®. The proposed techniques, after further tests, will be used in opportune hardware tests in the framework of existing projects such

as the Italian research project SUONO (*Safe Underwater Operations iN Oceans*) and the European research project ARROWS, coordinated by the MDM Lab of the University of Florence, to obtain initial experimental results [20].

9.2 I-AUV Multibody Modelling

9.2.1 I-AUV Description

The vehicle possesses 6 DOFs and is equipped with a manipulator arm, which is assumed to be a serial robot with 7 DOFs. On top of the wrist a 6-DOFs gripper is mounted: the latter has 3 fingers, each one composed of 2 phalanxes connected by rotational joints. The reference frames are shown in Fig. 9.1, linked to each rigid body and used to calculate the hydrodynamic terms.

9.2.2 I-AUV Kinematic and Dynamic Model

The analysis of the I-AUV model has been divided into two parts, separating the study of the vehicle model from the analysis of the manipulation system (i.e. the arm and the gripper). Geometrical and physical data have been set according to technical literature [21–23]. In this context, it is assumed that the gripper is rigidly connected to the robotic arm. The models are completely developed in Matlab-Simulink® environment.

SNAME notation has been used [1]; hence, the kinematic model of the AUV is defined in terms of η and ν vectors. η represents the position (η_1) and the orientation (η_2) written in the fixed reference frame $\langle n \rangle$; ν include the linear (ν_1) and the angular (ν_2) velocities described into the body reference frame $\langle b \rangle$. Both the fixed and the body reference frames use the NED directions.

The relations between $\dot{\eta}$ and ν can be written using the following expression:

$$\dot{\eta} = J_b^n(\eta_2) \nu, \tag{9.1}$$

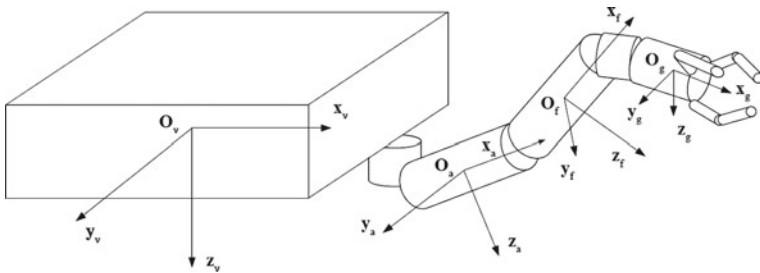


Fig. 9.1 Structure of the I-AUV, equipped with a robotic arm and a gripper

where

$$J_b^n(\boldsymbol{\eta}_2) = \begin{bmatrix} R_b^n(\boldsymbol{\eta}_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_b^n(\boldsymbol{\eta}_2) \end{bmatrix}. \quad (9.2)$$

$R_b^n(\boldsymbol{\eta}_2)$ is the rotation matrix between frame $\langle n \rangle$ and frame $\langle b \rangle$, and $T_b^n(\boldsymbol{\eta}_2)$ is the transformation matrix between angular velocity and the time derivative of Euler angles (and its form depends on the particular choice of Euler angles).

The dynamic model of the vehicle is defined as follows [1]:

$$M_{RB} \dot{\mathbf{v}} + C_{RB}(\mathbf{v}) \mathbf{v} = \boldsymbol{\tau}_H(\mathbf{v}, \mathbf{v}_c) + \mathbf{g}(\boldsymbol{\eta}) + \boldsymbol{\tau}, \quad (9.3)$$

where M_{RB} and $C_{RB}(\mathbf{v})$ are, respectively, the mass matrix and the Coriolis and centrifugal effect matrix. $\mathbf{g}(\boldsymbol{\eta})$ and $\boldsymbol{\tau}$ are the contribution due to the gravity effects and the external forces and moments (due to the motors and to the interaction with the arm) applied to the vehicle as to the body reference frame $\langle b \rangle$. These contributes are referred to the rigid body characteristics. Instead, the hydrodynamic effects $\boldsymbol{\tau}_H(\mathbf{v}, \mathbf{v}_c)$ are partially decoupled from the dynamical equation in order to use the classical multibody modelling techniques. In particular, buoyancy and hydrodynamic effects are introduced into the model by means of generalized Lagrangian forces applied to each body of the multibody system. From the classic equation of motion for an underwater vehicle [1] and the absolute velocity \mathbf{v} written in the body reference frame $\mathbf{v} = \mathbf{v}_r + \mathbf{v}_c$ (where \mathbf{v}_r is the relative velocity and \mathbf{v}_c is the current velocity), the following expression for $\boldsymbol{\tau}_H(\mathbf{v}, \mathbf{v}_c)$ can be extracted:

$$\begin{aligned} \boldsymbol{\tau}_H = & -M_A \dot{\mathbf{v}}_r + C_{RB}(\mathbf{v}_r) \mathbf{v}_c + C_{RB}(\mathbf{v}_c) \mathbf{v}_r \\ & + C_{RB}(\mathbf{v}_c) \mathbf{v}_c - C_A(\mathbf{v}_r) \mathbf{v}_r - D(\mathbf{v}_r) \mathbf{v}_r. \end{aligned} \quad (9.4)$$

where M_A is the added mass matrix due to the fluid viscosity, C_A is the Coriolis and centrifugal added effects, and $D(\mathbf{v}_r)$ is the damping matrix. The interaction among the different system bodies and the fluid has been modelled by means of appropriate CFD analyses [1, 18, 24, 25]; the mathematical coupling between the multibody model and the fluid equations has been efficiently performed through the toolbox SimMechanics[®]. In particular, as regards the CFD analyses, ANSYS[®] CFX software has been used to evaluate the elements of the matrices M_A , C_A and $D(\mathbf{v}_r)$ for different values of \mathbf{v} , $\dot{\mathbf{v}}$ and for different motions of the vehicle.

It is useful to express the forces and moments by dimensionless coefficients, to use them in any condition of similarity. Concerning the effects of the hydrodynamic resistance, the elements of the damping matrix $D(\mathbf{v}_r)$ are evaluated expressing the forces and torques through the following six dimensionless parameters:

- front, lateral and vertical drag coefficients:

$$C_{D_x} = \frac{F_x}{\frac{1}{2} \rho_a A_f v^2} \quad C_{D_y} = \frac{F_y}{\frac{1}{2} \rho_a D L v^2} \quad C_{D_z} = \frac{F_z}{\frac{1}{2} \rho_a D L v^2}$$

- roll, pitch and yaw resistance coefficients:

$$C_{M_x} = \frac{M_x}{\frac{1}{2}\rho_a A_f D^3 \omega^2} \quad C_{M_y} = \frac{M_y}{\frac{1}{2}\rho_a D L^4 \omega^2} \quad C_{M_z} = \frac{M_z}{\frac{1}{2}\rho_a D L^4 \omega^2}$$

where the used symbols are: speed v , angular velocity ω , frontal area A_f , diameter D , fluid density ρ_a , length L , and F_i, M_i are force and moment (all of them acting on the \mathbf{i} axis). The geometrical and physical characteristics of the vehicle are based on the literature and are defined in Table 9.1.

The I-AUV is provided of a robotic arm with 7 DOFs installed on the bow of the vehicle, in the middle of its lower part. For the kinematic model of the robotic arm (Fig. 9.2), the joint coordinates $\mathbf{q} = [\theta_1 \theta_2 \dots \theta_7]^T$ and the end-effector pose $\mathbf{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T$ are defined. According to the Denavit–Hartenberg (D–H) approach, Table 9.2 collects the D–H parameters extracted for the arm. The main kinematic equations used to entirely describe the redundant manipulator are respectively, for the direct kinematics and for the differential kinematics:

$$T_7^0 = T_7^0(\mathbf{q}) \tag{9.5}$$

Table 9.1 I-AUV data

Characteristic	Value
Degrees of freedom	6
Length	0.8 m
Breadth	0.6 m
Height	0.4 m
Mass in air	100 kg

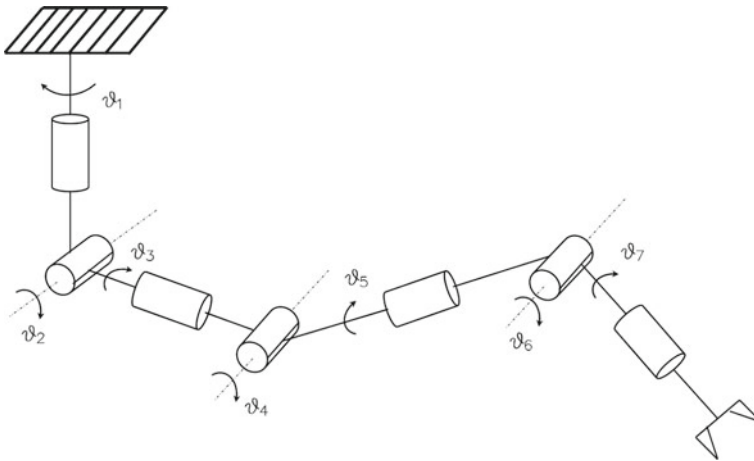


Fig. 9.2 Arm kinematic scheme

Table 9.2 Denavit–Hartenberg parameters of the arm

Link	a_i (m)	α_i (rad)	d_i (m)	ϑ_i (rad)
1	0	$\pi/2$	-0.05	θ_1
2	0	$\pi/2$	0	$\theta_2 + \frac{\pi}{2}$
3	0	$-\pi/2$	0.15	θ_3
4	0	$\pi/2$	0	θ_4
5	0	$-\pi/2$	0.15	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.05	θ_7

$$\mathbf{v}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = J\dot{\mathbf{q}}, \quad (9.6)$$

where $T_7^0 \in \mathbb{R}^{4 \times 4}$ is the homogeneous transformation matrix between the base reference frame $\langle 0 \rangle$ fixed to the AUV and the end-effector reference frame $\langle 7 \rangle$, $\mathbf{q} \in \mathbb{R}^{7 \times 1}$ is the vector of the joint variables, $\dot{\mathbf{p}}_e$ is the time derivative of the end-effector position and $\dot{\mathbf{q}}$ is the time derivative of the joint coordinates \mathbf{q} . The redundant DOFs are used to solve secondary tasks (e.g. the avoidance of the singularity or the minimization of the kinetic energy) [2].

The dynamic model of the robotic arm is simulated through the multibody techniques described before, in which each rigid body is modelled as follows:

$$M_l^i \dot{\mathbf{v}}_l^i + C_l^i(\mathbf{v}_l^i) \mathbf{v}_l^i = \boldsymbol{\tau}_H^i(\mathbf{v}_l^i, \mathbf{v}_{lC}^i) + \mathbf{g}^i(\boldsymbol{\eta}_l^i) + \boldsymbol{\tau}_l^i, \quad (9.7)$$

where M_l^i represents the mass matrix, $C_l^i(\mathbf{v}_l^i)$ is the Coriolis and centrifugal effect matrix of the i th link. $\mathbf{g}^i(\boldsymbol{\eta}_l^i)$ and $\boldsymbol{\tau}_l^i$ are respectively the contribution due to the gravity effects and the external forces (i.e. the torques of the actuators and the force arising from the interaction with the adjacent links) applied to the link (Table 9.3). The i th link characteristics define these contributes. As described before, the hydrodynamic effects $\boldsymbol{\tau}_H^i(\mathbf{v}_l^i, \mathbf{v}_{lC}^i)$ are partially decoupled from the dynamic equation in order to use the classical multibody techniques to solve the problem. In particular, these actions have been implemented in each body belonging to the I-AUV system (vehicle, links of the arm and gripper); the simulated effects include hydrostatic and hydrodynamics effects due to the added masses, drag and lift forces and buoyancy effects, implemented similarly to Eq. (9.4).

9.2.3 Gripper Multibody Model

A 3D model of the gripper is shown in Fig. 9.3. Each finger consists of two rotational joints connecting the hand to the first phalanx and the two phalanxes. A spherical

Table 9.3 Arm and forearm data

Parameter	Value
Length	0.15 m
Diameter	0.025 m
Mass	2.315 kg
Inertia (body frame)	
I_{xx}	$7.23 \times 10^{-4} \text{ kgm}^2$
I_{yy}	0.0047 kgm^2
I_{zz}	0.0047 kgm^2

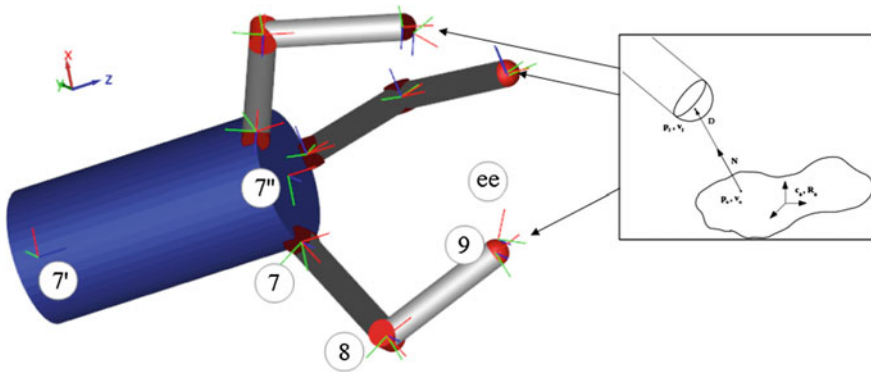


Fig. 9.3 Three-dimensional model of the gripper with the 3D contact model

tip is rigidly connected to the second phalanx. Using the D–H convention, the point where the first phalanx connects to the hand is the origin of reference frame 7 (for each finger), while the middle point of the finger is the origin of frame 8 and the end of the second phalanx is the origin of frame 9. Axis directions are chosen so that a positive value for D–H parameter θ corresponds to finger’s opening. The reference frame 7', visible in Fig. 9.3, which is the frame attached to the end effector of the arm, is tied to the frame 7'' (palm of the hand) by a constant transformation matrix; the same applies for the fingers’ frames 7 with respect to the palm frame. Finally, a frame is attached to the end of each fingertips, rotated by 45° with respect to the axis of the second phalanx. These frames are the end-effector (ee) frames of each finger. Mass and inertia values of the hand and the fingers are shown in Table 9.4.

Each finger is locally equivalent to a planar 2-DOFs manipulator, whose kinematics have been used; D–H parameters for one finger are reported in Table 9.5. The dynamic model of a finger in joint space is expressed by the well known relation

$$B(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + J_f^T \mathbf{h}_e = \boldsymbol{\tau}_f, \tag{9.8}$$

Table 9.4 Gripper data

Parameter	Hand	Phalanx
Length	0.1 m	0.05 m
Diameter	0.05 m	0.01 m
Mass	1.5433 kg	0.0309 kg
Inertia (body frame)		
I_{xx}	$4.82 \times 10^{-4} \text{ kgm}^2$	$3.87 \times 10^{-7} \text{ kgm}^2$
I_{yy}	0.0015 kgm^2	$6.62 \times 10^{-6} \text{ kgm}^2$
I_{zz}	0.0015 kgm^2	$6.62 \times 10^{-6} \text{ kgm}^2$

Table 9.5 Denavit–Hartenberg parameters of the finger

Link	a_i (m)	α_i (rad)	d_i (m)	ϑ_i (rad)
8	0.05	0	0	θ_8
9	0.05	0	0	θ_9

where B is the inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ include centrifugal, Coriolis and gravitational effects, J_f is the finger Jacobian matrix, $\mathbf{h}_e \in \mathbb{R}^{6 \times 1}$ is the vector of forces/torques due to interactions with the environment and $\boldsymbol{\tau}_f$ are the joint torques. As for the vehicle and the arm, multibody modelling techniques have been used for the gripper.¹

Contact model

This section describes the model that simulates the contact between the spherical tips of the gripper’s fingers and the object to be manipulated. The contact model has the following features [26]:

- contact point: it is assumed that there is a single contact point;
- hard finger contact: tangential forces arise due to friction;
- 3D model: even if the model has been created to govern the contact of the specific test case, its geometrical background is easily adaptable to different cases; thus, it is a complete three-dimensional model.

Figure 9.3 presents the notation used in the model: \mathbf{c}_o, R_o denote the pose of the object in the inertial frame; $\mathbf{p}_f, \mathbf{p}_o$ and $\mathbf{v}_f, \mathbf{v}_o$ represent the position and the velocity of the contact points on the fingertip and on the object, respectively. $\mathbf{D} = \mathbf{p}_f - \mathbf{p}_o$ is the distance between the contact points, and \mathbf{N} is the contact normal (pointing outwards the object); finally, let $\mathbf{s} = \mathbf{v}_o - \mathbf{v}_f$ denote the sliding between the surfaces in contact (i.e. the difference between the contact points velocities).

The algorithm can be divided into two steps: firstly the position of the contact points is determined, along with their (vectorial) distance \mathbf{D} ; if there is penetration

¹In Fig. 9.3 two reference frames are visible on every finger joint; one is attached to the link, while the other shows the orientation when joint coordinate θ is zero. If all joints coordinates are zero, fingers are stretched.

Table 9.6 Contact model parameters

Parameter	Value
Elastic coefficient	$k_n = 10000 \text{ kg/s}^2$
Damping coefficient	$c_n = 100 \text{ kg/s}$
Static friction coefficient	$\mu_s = 0.5$
Kinetic friction coefficient	$\mu_k = 0.3$
Friction curve parameter	$k = 5$

(i.e. $\rho = \mathbf{D}^T \mathbf{N}$ becomes negative), the contact forces are computed according to the model presented in the following section [27].

Hard finger contact has been considered: three components of force are transmitted at contact; one is normal to the surface, while the others are frictional (tangential) forces. Normal force follows a spring-damper model: let

$$s_N = \mathbf{s}^T \mathbf{N} \quad (9.9)$$

denote the value of the component of the sliding vector along \mathbf{N} ; then, normal contact force on the object is given by

$$\mathbf{f}_N = (k_n \rho - c_n s_N) \mathbf{N}, \quad (9.10)$$

for $k_n > 0$, $c_n > 0$. The tangential force is composed of static and kinetic friction; the friction coefficient μ follows the law:

$$\mu(\|\mathbf{s}_t\|) = \mu_k + (\mu_s - \mu_k) e^{-k\|\mathbf{s}_t\|}, \quad (9.11)$$

where $\mathbf{s}_t = \mathbf{s} - s_N \mathbf{N}$ is the tangential sliding, $\mu_s > \mu_k$ are the static and kinetic friction coefficients and $k > 0$ is a tunable parameter. Tangential force exerted on the object is then:

$$\mathbf{f}_t = -\mu \|\mathbf{f}_N\| \frac{\mathbf{s}_t}{\|\mathbf{s}_t\|}. \quad (9.12)$$

Finite slope can be assumed for $\mu(\|\mathbf{s}_t\|)$ for small sliding values, to avoid chattering problems during simulations. As a contact between two steel surfaces has been considered, the chosen contact parameters [28] are reported in Table 9.6.

9.2.4 Camera Model

During the grasp planning phase a pose estimation algorithm based on optical sensors (cameras) has been used, which gives the position of the center of mass of the object \mathbf{c}_o and its orientation R_o with respect to the arm; thus, it is necessary to give a

mathematical model of such sensors. In this context, the pinhole model has been used: this model offers a high computational efficiency [29].

9.3 I-AUV Control

The control system has to make sure that the system can autonomously reach the object to be manipulated and execute the planned task on it.

It has been assumed that all the DOFs of the system are controlled. Actuating forces and torques and their variations have been limited by means of saturations and rate limiters, to simulate the presence of a real actuator [1, 30–32]. At the same time, navigation sensors have been modelled [33].

I-AUVs' control techniques can be divided into two categories [8]. A first set simultaneously controls the vehicle and the manipulation system [14], subjugating the first to the latter. The second set of techniques makes use of a decoupled approach [15, 34]: the arm manipulates the object while the vehicle tracks its own reference trajectory; the effects of one subsystem on the other are considered as disturbances.

Because of its simplicity and robustness with respect to unknown parameters of the system, the second approach has been chosen. Moreover, exploiting the kinematics of the gripper, the control of the fingers has been further decoupled from the control of the arm: once the desired pose for the fingertips is known, it is indeed possible to univocally determine, from purely geometrical considerations, the desired pose of the arm's end effector.

A block diagram of the control architecture is shown in Fig. 9.4. Filled lines represents physical interactions, while dashed lines stand for functional dependence. The global trajectories of the system's components must permit the manipulation of the object; the dotted line connecting the vehicle's and the manipulator's trajectory planning blocks indicates that, even if the two references are virtually independent because of the adopted decoupled strategy, the AUV's trajectory must allow the arm to reach the object.

Indeed, the whole control system follows a "backward strategy": during manipulation, the contact points on the object are computed by the grasp planning algorithms; then, a suitable smooth trajectory that takes the fingertips on such points is generated. The positional control of the fingers is in charge of following this trajectory closely. The geometrical decoupling algorithm allows the computation of the reference trajectory of the arm, which is kinematically controlled. Finally, an admissible reference trajectory for the AUV is generated, and a SISO PID control is applied to each DOF of the vehicle [1].

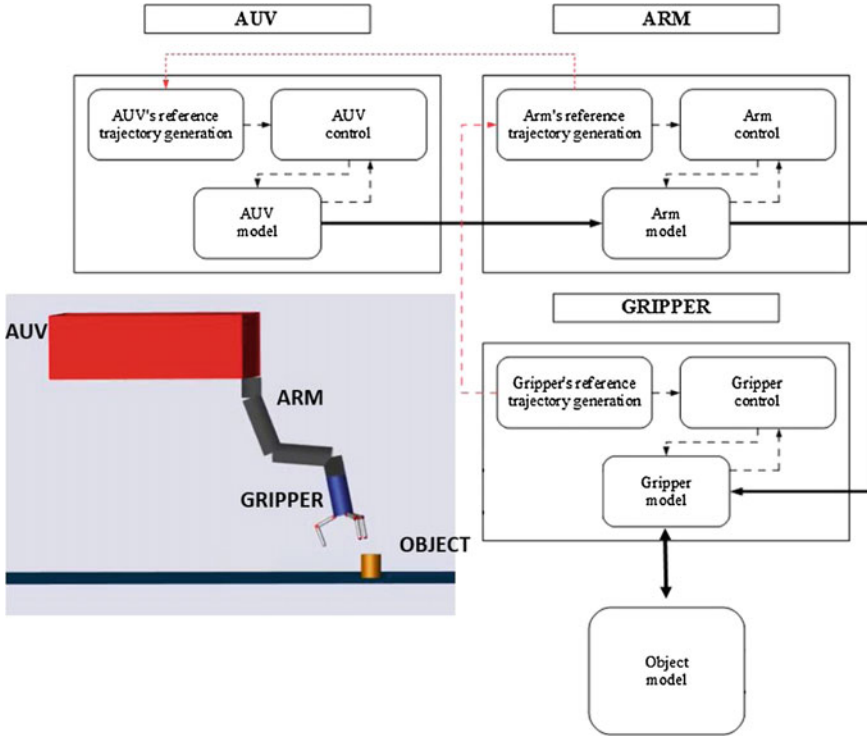


Fig. 9.4 Control architecture of the I-AUV

9.3.1 Trajectory Generation of the Manipulation System

The contact points on the object surface constitute the desired values of the reference trajectory; the desired orientation is chosen so that the approach axis of a fingertip frame points inwards the object [35]. The reference trajectory is generated as convex combination of initial and final pose. Let us consider position trajectory generation first: let x denote the generic (scalar) position variable; its desired trajectory is then chosen as

$$x(t) = \lambda(t)x_i + (1 - \lambda(t))x_d, \tag{9.13}$$

where x_i and x_d represent the initial and the desired value of x and $\lambda(t)$ is a parameter that continuously (and with continuous derivatives) varies from 1 to 0. This ensures a smooth transition between the initial and the desired values. The continuity of the trajectory is maintained even if the desired value x_d varies with time.

For orientation trajectory generation, initial and desired orientations are expressed as unit quaternions \mathbf{q}_i and \mathbf{q}_d [36]; then, Spherical Linear interpolation (SLERP) is applied, in order to compute a constant angular velocity rotation:

$$\mathbf{q}(t) = \frac{\sin(1 - \lambda(t))\theta}{\sin(\theta)} \mathbf{q}_i + \frac{\sin(t\theta)}{\sin(\theta)} \mathbf{q}_d, \quad \theta = \cos^{-1}(\mathbf{q}_i^T \mathbf{q}_d). \quad (9.14)$$

While the I-AUV is patrolling, searching for the object to be manipulated, the arm is kept at rest position. It is assumed that a camera is mounted on the palm of the gripper. Supposing that the shape of the object is known a priori, as soon as the object enters the field of view of the camera the POSIT algorithm (*Pose from Orthography and Scaling with Iterations* [37]) can be executed to obtain an estimation of the pose of the object itself. Then, the arm's reference trajectory is changed so as to ensure that the object is kept inside the field of view of the camera all the time. During manipulation, the arm's reference trajectory is obtained by means of a geometrical algorithm that determines the pose of the end effector of the arm from the knowledge of the fingertips pose. The solution is unique, and the solving process is composed of just a few steps. To overcome the problem of obtaining exact values for the hydrodynamic coefficients of the arm (whereof only an estimation is available), a kinematic control has been preferred to a dynamic control strategy. A Closed Loop Inverse Kinematic Control (CLIK) has been chosen [38], exploiting the arm's redundancy to keep the manipulator far from singularities.

9.3.2 Trajectory Generation of the AUV

As the definition of dynamic manipulation implies, the AUV never stops during the execution of the task; furthermore, the vehicle must constantly keep the object inside the arm and gripper workspace so as the manipulation takes place.

The AUV is controlled by means of a decoupled PID strategy: six PID controllers have been used, one for each degree of freedom of the vehicle [1]. The control law is:

$$\mathbf{u}_{AUV}^{th} = H^\dagger \mathbf{u}_{AUV}, \quad (9.15)$$

where

$$\mathbf{u}_{AUV} = J_b^n \boldsymbol{\tau}_{PID} + \mathbf{g}(\boldsymbol{\eta}), \quad (9.16)$$

being $\boldsymbol{\tau}_{PID}$ a 6×1 vector of PID action on the pose error \mathbf{e}_η . J_b^n is the AUV Jacobian matrix, $\mathbf{g}(\boldsymbol{\eta})$ a term of gravity compensation, and H^\dagger is a generalized pseudo-inverse of the propeller matrix H [1], which maps the vector of thruster forces \mathbf{S} into the vector of forces/torques acting on the vehicle:

$$\boldsymbol{\tau} = H\mathbf{S}. \quad (9.17)$$

In the considered case, $\mathbf{S} = [S_1 \ S_2 \ S_3 \ S_4 \ S_5 \ S_6]^T$; hence, the propeller matrix is square and it has the form:

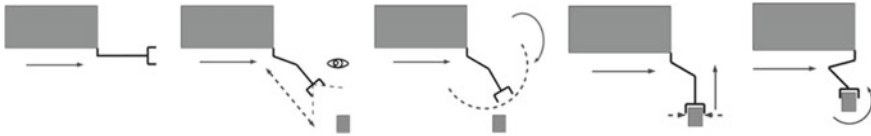


Fig. 9.5 Description of the performed manipulation task

Table 9.7 Simulation data

Test case parameters			Cylinder parameters	
AUV cruising speed	Acceleration time	Acceleration profile	Radius	2.5 cm
0.1 m/s	5 s	Trapezoidal	Length	6 cm
0.2 m/s	5 s	Trapezoidal	Density	7860 kg/m ³
0.25 m/s	5 s	Trapezoidal	Mass	926 g

$$H = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 \\ \mathbf{P}_{m1} \times \mathbf{v}_1 & \mathbf{P}_{m2} \times \mathbf{v}_2 & \mathbf{P}_{m3} \times \mathbf{v}_3 & \mathbf{P}_{m4} \times \mathbf{v}_4 & \mathbf{P}_{m5} \times \mathbf{v}_5 & \mathbf{P}_{m6} \times \mathbf{v}_6 \end{bmatrix}, \quad (9.18)$$

where \mathbf{P}_{mi} for $i = 1-6$ is the propeller position vector in body reference frame and \mathbf{v}_i for $i = 1-6$ is the unitary vector of the thrust direction.

9.4 Numerical Simulations

In this section, the dynamic behaviour of the whole I-AUV system has been simulated during the execution of a predefined dynamic manipulation task (whose scheme is presented in Fig. 9.5).

The task consists of grasping a cylindrical object lying on the sea floor, and it is composed of the following steps:

- the I-AUV, starting from rest, accelerates for 5 s until steady-state speed is reached;
- as soon as the cylinder enters the field of view of a camera mounted on the palm of the gripper (eye-in-hand configuration), the reference trajectory is changed so as to align the camera focal axis with the line connecting the palm of the gripper to the position of the center of mass of the object (estimated by the POSIT algorithm); this way, the cylinder is kept inside the field of view of the camera all the time;
- when a threshold distance between the gripper and the object is reached, the arm is lowered and the gripper grasps the cylinder;
- the cylinder is lifted and carried as the arm reaches a final “rest” configuration (i.e. the cylinder carried vertically under the bow of the vehicle).

It is worth noting that the AUV never stops during the execution of the task, which is a fundamental requirement for a correct dynamic manipulation task.

For the ease of reading, reported data (unless stated otherwise) are expressed in an inertial reference frame whose x axis is aligned with the forward direction of the vehicle, z axis is directed toward the sea surface, and y axis forms a right-handed coordinate system. Without loss of generality, the origin of this frame is chosen coincident with the initial position of the center of mass of the AUV. The object lies on the sea floor at about -0.6 m under the vehicle (initial depth). The described task has been simulated using Matlab-Simulink[®] software. A fixed step solver (*ODE5-Dormand-Prince*) has been chosen to increase affinity with real hardware.

Three simulations have been carried out, at different vehicle's speeds: 0.1, 0.2 and 0.25 m/s; this makes possible the analysis of the effect of increasing speed on the performances of the control system. Table 9.7 summarizes the main parameters of the simulations and the cylinder properties. At higher speed, the I-AUV fails to execute the task correctly; however, this is not due to a control architecture fault, but to the physical length of the links of the arm which impose a limit on the window of time allowed for manipulation.

Figure 9.6 reports the three-dimensional trajectory of the vehicle, the gripper, the fingertips and the object obtained during the fastest simulation.

Figure 9.7 shows the behaviour of the AUV during the three simulations: the error along the direction of forward motion is very small, its value increasing with the speed of the vehicle: this is because the AUV always accelerates for 5 s, thus higher steady-state speed equals higher acceleration values, which decrease the performance of the PID controller. Y-motion, roll and yaw angular errors are neglectable. Initial errors on the z -axis motion and on the pitch angle, not affected by speed, are due to the total buoyancy of the system: while the vehicle has positive buoyancy (1%), the arm and the gripper tend to sink the I-AUV; in addition, since the manipulation system is mounted centrally on the front side of the AUV, it has the effect of leaning the vehicle forward. However, these errors are kept small and rejected in time.

Arm position errors, visible in Fig. 9.8, are expressed in a local coordinate system whose origin coincides with the shoulder of the manipulator and whose initial orientation is the same as the inertial frame. Aside from the initial error (due to gravity), Fig. 9.8 shows that the error increases as the AUV moves faster; however, it is kept small during the execution of the manipulation task. Fingertips position errors are very similar to the arms ones, thus they are not reported.

Figure 9.9 shows the position of the center of mass of the cylinder on each axis. Plots have been divided according to x , y and z coordinates. It is clearly visible the time when the object is grabbed and then lifted (motion on the x and z axis, respectively); transversal motion is neglectable (Table 9.8).

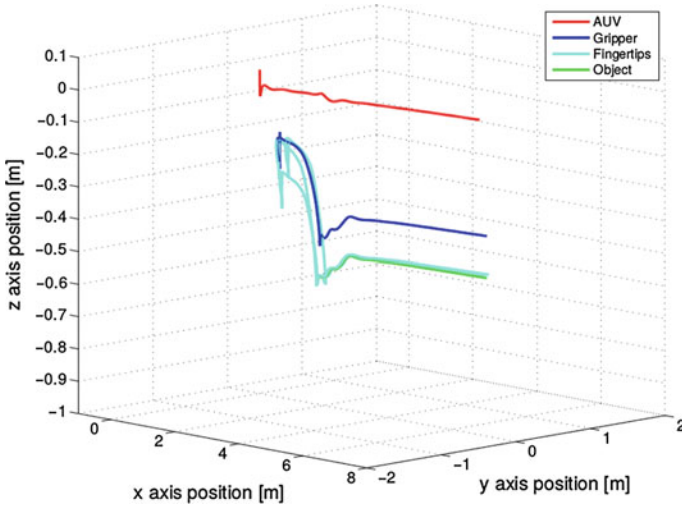


Fig. 9.6 3D trajectory

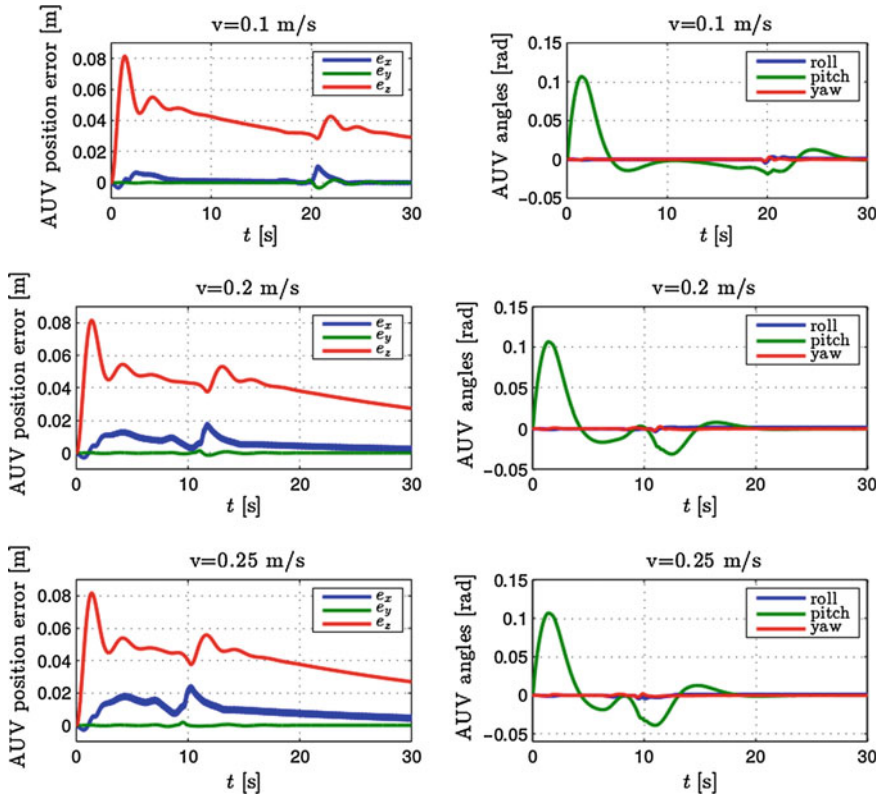


Fig. 9.7 Pose error of the AUV

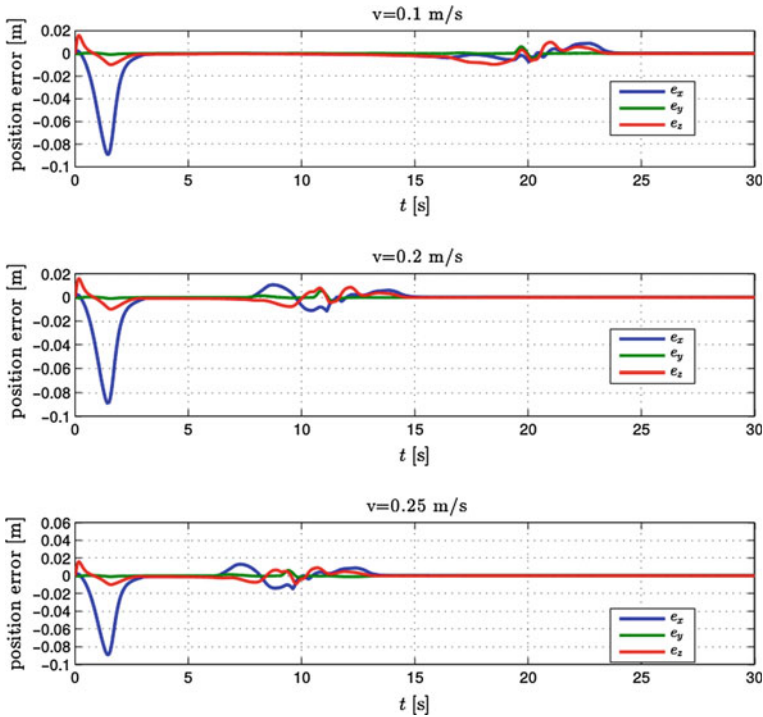


Fig. 9.8 Position error of the arm

9.5 Conclusion

This paper describes a detailed multibody model of an I-AUV to study a suitable control architecture for manipulation tasks. In particular, to better analyse the effectiveness of the multibody models, the most challenging autonomous manipulation (dynamic manipulation) has been considered. Dynamic manipulation denotes manipulation tasks executed while the vehicle maintains relevant velocities, further complicating the execution of the mission due to the dynamic interaction between the AUV and the manipulator. A complete multibody model of the I-AUV system has been derived, including interaction with the fluid and contact with the object to be manipulated. The I-AUV is controlled by means of a decoupled vehicle-manipulator strategy, further decoupling the control of the gripper exploiting the hand kinematics.

Different relative speeds between the I-AUV and the object, in the same simulation scenario, have been simulated with satisfying results, showing how the developed multibody models and the adopted strategy allow the execution of the task. As concerns future investigations, different simulation scenarios are required to establish the maximum velocity that can be maintained during the manipulation phase. Further improvements are scheduled, with special attention given to data acquisition and

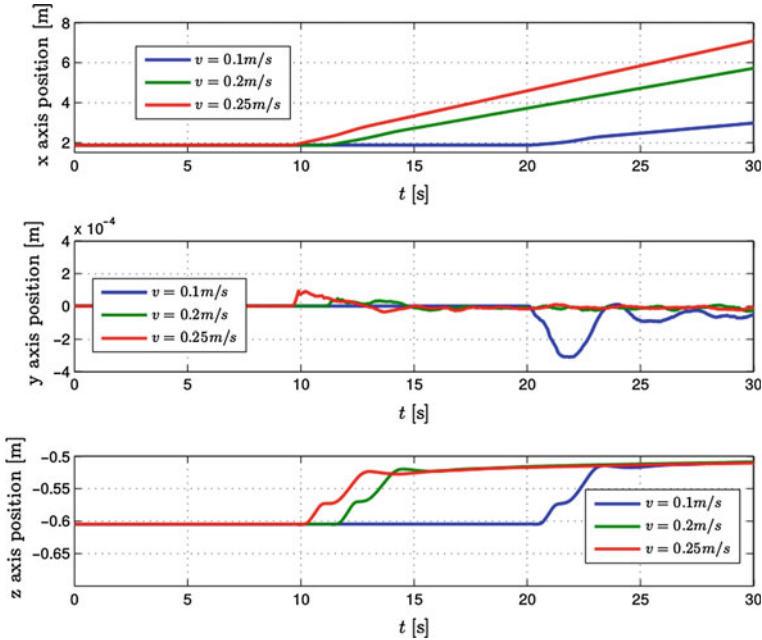


Fig. 9.9 Cylinder position

Table 9.8 Minimum AUV speed

Nominal speed	Minimum speed
0.1 m/s	0.08 m/s
0.2 m/s	0.17 m/s
0.25 m/s	0.22 m/s

to autonomous calculations, before the application of the proposed strategy in the framework of the Italian project SUONO and of the FP7 European project ARROWS.

Acknowledgments This work has been partially supported by the European ARROWS project, that has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 308724.

References

1. Fossen TI (1994) Guidance and control of ocean vehicles, 1st edn. Wiley, Chichester
2. Siciliano B, Khatib O (2008) Handbook of Robotics. Springer Handbooks, Napoli and Stanford
3. Murray RM, Li Z, Sastry S (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton

4. Sanz PJ, Ridao P, Oliver G et al (2010) TRIDENT: a framework for autonomous underwater intervention missions with dexterous manipulation capabilities. In: The 7th symposium on intelligent autonomous vehicles. Lecce, Italy
5. Hildebrandt M, Christensen L, Kerdels J, Albiez J, Kirchner F (2009) Realtime motion compensation for ROV-based tele-operated underwater manipulators. In: OCEANS 2009—EUROPE, vol 1, pp 1–6, 11–14 May 2009, doi:[10.1109/OCEANSE.2009.5278241](https://doi.org/10.1109/OCEANSE.2009.5278241)
6. Christensen L, Kampmann P, Hildebrandt M, Albiez J, Kirchner F (2009) Hardware ROV simulation facility for the evaluation of novel underwater manipulation techniques. In: OCEANS 2009—EUROPE, pp 1–8, 11–14 May 2009, doi:[10.1109/OCEANSE.2009.5278249](https://doi.org/10.1109/OCEANSE.2009.5278249)
7. Marani G, Choi SK, Yuh J (2008) Underwater autonomous manipulation for intervention missions AUVs. *Ocean Eng* 36(1):15–23
8. Antonelli G (2006) *Underwater robots*, springer tracts in advanced robotics, 2nd edn. Springer, Heidelberg
9. Ridao P, Carreras M, Ribas D, Sanz PJ, Oliver G (2014) Intervention AUVs: the next challenge. In: World IFAC, Vol 19, Part 1, pp 12146–12159, Cape Town, SA, Aug 24–29
10. Eielsen AA (2012) *Autonomous underwater robotics: state of the art and future challenges*, Phd Trial Lecture, NTNU, Trondheim, Nov 26
11. Breivik M, Fossen TI (2005) Guidance-based path following for autonomous underwater vehicles. In: Proceedings of the OCEANS'05, vol 3, Washington, DC, USA, pp 2807–2814
12. Lane DM, Davies JBC, Casalino G et al (1997) AMADEUS: advanced manipulation for deep underwater sampling. In: IEEE Robotics and Automation Magazine, pp 34–45, Universitat de Barcelona
13. Caffaz A, Caiti A, Casalino G, Turetta A (2010) The hybrid glider-AUV folaga. *IEEE Robot Autom Mag* 17(1):31–44
14. McLain TW, Rock SM, Lee MJ (1996) Coordinated control of an underwater robotic system. In: Video proceedings of the IEEE international conference on robotics and automation, Minneapolis, pp 213–232, MN
15. Kim J, Chung WK, Yuh J (2003) Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems. In: IEEE/RSJ International conference on intelligent robots and systems, pp 577–582, Las Vegas, NV
16. Simetti E, Casalino G, Torelli S, Sperinde A, Turetta A (2014) Floating underwater manipulation: developed control methodology and experimental validation within the TRIDENT project. *J Field Robot* 31(3):364–385
17. PANDORA FP7 project <http://persistentautonomy.com/>
18. Fox RW, McDonald AT (2004) *Introduction to fluid mechanics*, 6th edn. Wiley, USA
19. Furferi R, Governi L, Volpe Y, Carfagni M (2013) Design and assessment of a machine vision system for automatic vehicle wheel alignment. *Int J Adv Robot Syst* 10(242):1–16
20. Allotta B, Costanzi R, Ridolfi A et al (2015) The ARROWS project: adapting and developing robotics technologies for underwater archaeology. In: Proceedings of IFAC workshop on navigation guidance and control of underwater vehicles (NGCUV 2015). Girona, Spain
21. Autonomous Undersea Vehicle Applications Center, <http://auvac.org>
22. *Remotely operated vehicles of the world*, 7th ed. Clarkson Research Services Ltd. (2006/2007)
23. Allotta B, Conti R, Costanzi R, Giardi F, Meli E, Ridolfi A (2013) Modelling and control of an autonomous underwater vehicle for mobile manipulation. In: Proceedings of multibody dynamics 2013 congress (ECCOMAS), Zagreb, Croatia
24. Lee SK (2006) CFD simulation for propeller four-quadrant flows, ABS Technical Papers
25. Allotta B, Costanzi R, Monni N, Pugi L, Ridolfi A, Vettori G (2012) Design and simulation of an autonomous underwater vehicle. In: Proceedings of the European congress on computational methods in applied sciences and engineering (ECCOMAS), Vienna, Austria, Sept 10–14
26. Wriggers P (2006) *Computational contact mechanics*, 2nd edn. Wiley, New York
27. Wittenburg J (2008) *Dynamics of multibody systems*. Springer, Berlin
28. Cheli F, Pennestri E (2006) *Kinematics and dynamics of multibody systems*. Casa Editrice Ambrosiana, Milano

29. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
30. Carlton J (2007) Marine propellers and propulsion, 2nd edn. Elsevier, Amsterdam
31. Allotta B, Bartolini F, Costanzi R, Monni N, Pugi L, Ridolfi A (2014) Preliminary design and fast prototyping of an autonomous underwater vehicle propulsion system. In: Proceedings of the institution of mechanical engineers, part m: journal of engineering for the maritime environment, Jan. 27, doi:[10.1177/1475090213514040](https://doi.org/10.1177/1475090213514040)
32. Pivano L, Johansen TA, Smogeli ØN (2009) A four-quadrant thrust estimation scheme for marine propellers: theory and experiments. *IEEE Trans Control Syst Technol* 17(1):215–226
33. Conti R, Meli E, Ridolfi A, Allotta B (2015) An innovative decentralized strategy for I-AUVs cooperative manipulation tasks. *Robot Auton Syst* 72(C):261–276
34. Chang K-S, Khatib O (1995) Manipulator control at kinematic singularities: a dynamically consistent strategy. In: Proceedings of the IEEE/RSJ international conference on intelligent robot and systems 95 “Human robot interaction and cooperative robots”, vol 3, pp 84–88
35. Lippiello V, Ruggiero F, Siciliano B, Villani L (2013) Visual grasp planning for unknown objects using a multifingered robotic hand. In: *IEEE/ASME Trans Mechatron* 8(3):1050–1059
36. Yuan JS-C (1988) Closed-loop manipulator control using quaternion feedback. *IEEE J Robot Autom* 4(4):434–440
37. DeMenthon Daniel F, Davis Larry S (1995) Model-based object pose in 25 lines of code. *Int J Comput Vis* 15(1–2):123–141
38. Chiacchio P, Chiaverini S, Villani L, Siciliano B (1991) Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *Int J Robot Res* 10(4):410–425

Chapter 10

Development of a Musculotendon Model Within the Framework of Multibody Systems Dynamics

Ana R. Oliveira, Sérgio B. Gonçalves, Mamede de Carvalho and Miguel T. Silva

Abstract Human movement is the result of a complex and synergistic interaction between the musculoskeletal and the central nervous system. As result, muscles contract coordinately to produce forces that are transmitted by tendons to the skeletal system, causing its movement or keeping its pose. Often neglected in current muscle models, the elastic properties of tendons play a significant role in the dynamic interaction between the muscular and skeletal systems, influencing the force transmission, energy storage and transfer, and joint control. The aim of this work is to present in detail the necessary steps to incorporate a musculotendon model in the framework of a multibody systems dynamics formulation. A methodology to compute the musculotendon forces and activations is presented based on the use of a Hill-type muscle model assembled in series with a spring-like element defined according to the elastic properties of the tendon. The proposed methodology can be applied, without significant changes, to both inverse and forward dynamic analyses of biomechanical systems. Three daily activities with different levels of musculotendon recruitment are analyzed from an inverse dynamics perspective. The selected activities are walking, running and jumping. The movement data characterizing these activities were acquired experimentally in a movement laboratory. A 3D biomechanical model of the human body, described with natural coordinates and encompassing 43 musculotendon actuators per leg, is proposed to assess the performance of the presented musculotendon model and of its incorporation on the referred multibody dynamics framework. The influence of the introduction of a compliant tendon model on the produced muscle forces and activation patterns is analyzed in face of those same results produced by the same biomechanical model defined with infinitely stiff

A.R. Oliveira · S.B. Gonçalves · M.T. Silva (✉)
IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
e-mail: MiguelSilva@tecnico.ulisboa.pt

A.R. Oliveira
e-mail: Ana.r.sousa.de.oliveira@tecnico.ulisboa.pt

S.B. Gonçalves
e-mail: Sergio.goncalves@tecnico.ulisboa.pt

M. de Carvalho
Faculty of Medicine, Universidade de Lisboa, Lisbon, Portugal
e-mail: mamedevalves@medicina.ulisboa.pt

(or rigid) tendons. Results revealed that the introduction of the tendon model allows muscles to work, predominantly, on their optimal configuration as the dynamic equilibrium generated between muscle and tendon prevents the muscle from support all musculotendon deformation. This not only reduces the activations needed to perform the required contractile forces but it also considerably prevents the development of non-physiological passive forces.

Keywords Musculotendon model · Tendon model · Hill-muscle model · Contraction dynamics · Multibody dynamics · Motion analysis

10.1 Introduction

Last decades have been marked by a large number of works researching on the nature of human movement. This growing interest is related with the positive impact that such knowledge can have in different research and application areas from which biomechanics, health and sports sciences, and robotics are here highlighted. Understanding the specificities of the normal and abnormal movement is, thus, essential to develop novel solutions that allow the improvement of human performance and the quality of life in the presence of impairment.

Human movement is the result of an intricate process involving the musculoskeletal and the central nervous system (CNS). A complex pattern of electric signals—neural stimuli—generated at CNS level, results in a synergistic contraction of muscle fibers that induces forces, which are transmitted by tendons to the skeletal system, causing movement or maintaining pose. As such, the study of musculotendon activity is of great interest for the scientific and medical community alike, as it allows for better understanding the contribution of specific musculotendon groups to a given movement [1], thus providing insight to the design of customized orthotic and prosthetic solutions (e.g. functional neuromuscular stimulation systems [2]), or other assistive devices.

Tendons mediate the interaction between muscular and skeletal systems, influencing the force transmission, energy storage and transfer, joint control, movement dynamics and accuracy [3]. Moreover, due to their intrinsic viscoelastic properties, tendons have an important role in the prevention of muscle injuries, behaving as a mechanical buffer that avoids excessive elongations of muscle fibers [4]. Hence, the development of non-invasive computer methods based on musculoskeletal modelling in which tendons are also considered is of particular interest as these methodologies enable the analysis of the dynamics underlying these interactions and their influence on movement.

Several mathematical models have been presented to characterize the dynamics of muscle tissue, standing out the ones proposed by Huxley [5] and Hill [6]. Based on the physiological structure of muscles, Huxley-type models estimate the muscles forces considering the cross-bridges between the actin and myosin filaments. Although these models present a strong physiological component, each muscle is

ruled by a set of differential equations, one per cross-bridge, substantially increasing the computational complexity required to solve the problem [7]. On the other hand, Hill-type muscle models focus on the mechanical behavior of the muscle, considering, in most of cases, only one differential equation per muscle. As result, the computational effort is not as high as in the Huxley model, making this type of models suitable for analyze or simulate biomechanical models described with large number of musculotendon actuators [7].

Regardless of the large amount of studies utilizing computational muscle models, few discuss tendon action. Some works consider tendon as an infinitely stiff (or rigid) element, with a fixed length, being contraction dynamics supported solely by the muscle actuator. Regarding this approach, Thelen et al. refer that this fact could explain the non-physiological muscle strains and forces observed in some of these works [4]. When compliant (or non-rigid) tendons are considered, almost all methodologies found in literature implement or adapt the model proposed by Zajac [8], in which pure elastic properties are considered for the tendon and musculotendon dynamics is described by a first order differential equation. There, a detailed analysis of the mechanical properties of the tendon is presented, discussing the relations between tendon force, tendon stress and tendon strain. The referred work addresses also a theoretical base to develop a dimensionless linear model that allows for the computation of the principal muscle- and tendon-related parameters, considering the use of a Hill-type muscle model to describe muscle contraction dynamics. Since then, this model has become the standard methodology for the study of musculotendon dynamics, being present in most studies that consider compliant tendons [4, 9–14].

Millard et al. analyzed the forces produced by three different musculotendon models: one considering rigid tendons and two describing tendons as compliant elements (the equilibrium model and the damped equilibrium model) [10]. Results showed that the compliant models presented similar responses, with force patterns denoting higher physiological meaning than those obtained with the rigid tendon model. Additional analyses on the computational effort concluded that although slower than the rigid model, the damped equilibrium model was in fact faster than the (solely elastic) equilibrium model, particularly in the presence of low activation levels.

It is therefore the objective of this work to present, in a comprehensive manner, a methodology to incorporate a musculotendon model that takes into account the influence of the tendon on muscle contraction in the framework of a multibody systems dynamics formulation. The proposed methodology can be applied both in forward and inverse dynamics analyses of large biomechanical systems. The musculotendon unit is based on the theoretical model proposed by Zajac [8] and is described by a Hill-type muscle model assembled in series with an elastic element representing the mechanical properties and behavior of the tendon.

The proposed methodology was implemented in APOLLO, a three-dimensional multibody dynamics analysis software with natural coordinates developed in-house [15]. The performance of the methodology is assessed by applying it to the analysis of three daily activities with different levels of tendon recruitment: walking, running and jumping. For this purpose a biomechanical model with 43 musculotendon units

per leg is utilized. The influence of the tendon on muscle contraction dynamics is also analyzed by comparing the obtained results (activations, musculotendon forces and lengths) with a musculotendon model in which the tendon is modelled assuming rigid properties, i.e. the length of the tendon is considered constant and equal to its respective slack length [16]. Discussion and conclusions are presented in face of the obtained results.

10.2 Methodology

10.2.1 Musculotendon Model

Physiological muscle contraction begins with the generation of a neural stimulus by the CNS that travels through the motor neuron until it reaches the neuromuscular junction of the target muscle fiber. When the neural stimulus reaches the axon terminal, a group of neurotransmitters—acetylcholine—is released resulting in the generation of an action potential on the muscle cell and, consequently, inducing its contraction on a process that is referred to as activation dynamics. The muscle contraction process itself can be explained by the sliding-filament theory, where the actin filaments are pulled by the myosin filaments through cross-bridges formed between the actin binding sites and the myosin heads [17]. The force developed by the muscle is a function of the amount of overlap of these two protein filaments in a process that is referred to as contraction dynamics [18].

The computational musculotendon model should also contemplate these complex dynamic processes as these confer extended physiological relevance to the model and results. Hence, the computational model should include an activation dynamics block, which describes the transformation of a neural excitation $u^M(t)$ in the respective muscle activation $a^M(t)$ and a contraction dynamics block that corresponds to the conversion of the muscle activation $a^M(t)$ into musculotendon force (F^{MT}), as schematically represented in Fig. 10.1 [1, 7].

From the computational point of view the activation block models the physiological delay that exists between the neural stimulus $u^M(t)$ and the generation of the muscle activation $a^M(t)$, also referred to as electromechanical delay (EMD) [7]. This delay is the result of the time that takes the propagation of the neural signal through the neuron and the duration of the biochemical cascade of events resulting in the excitation of the muscle fiber. This physiological process can be modeled as a first order system but the solution of such systems, particularly in inverse dynamic analysis, requires the use of an optimization strategy different from the one adopted in this work, i.e., the solution of the inverse dynamics problem considering the EMD requires the use of a dynamic optimization strategy that globally optimizes for multiple points in time in order to encompass the effect of the EMD in the solution. In this work, a static optimization strategy was adopted instead and the optimization is carried out locally for a single point in time, which allows the

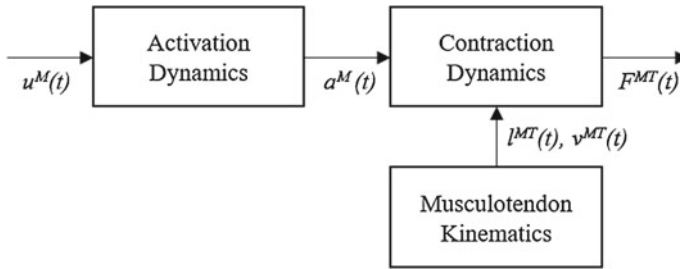


Fig. 10.1 Musculotendon dynamics: The activation dynamics block governs the transformation of the neural excitation $u^M(t)$, produced by CNS, into an activation state of the muscle tissue $a^M(t)$, while the conversion of $a^M(t)$ into musculotendon force $F^{MT}(t)$ is governed by the contraction dynamics block considering the actual state (length and contraction velocity) of the musculotendon actuator

simulation of contraction dynamics but excludes the inclusion of activation dynamics and the simulation of the EMD. Examples of classical muscle activation models can be consulted in [7, 8]. Other works addressing artificial activation models can also be found in literature. These models are of particular interest as they allow to understand the response of the muscle to external signals, such as functional electrical stimulation (FES) profiles [19]. Such knowledge enables the design of biomedical solutions customized for a particular subject and neuromuscular disorder [2].

The contraction dynamics block models the process of generating an effective musculotendon force (F^{MT}), considering the state of the muscle (i.e., its length and contraction velocity) and a given activation level $a^M(t)$. Depending on the type of dynamic analysis, the activation level can be a known value, prescribed as input to the simulation in the case of forward dynamic analysis, or an unknown value to determine as the result of an inverse dynamic analysis. Hill-type muscle models are a common approach to model contraction dynamics since they allow the characterization of the dynamic behavior of the muscle tissue with only one differential equation per muscle, this way reducing the computational effort required to solve the problem and enabling large-scale muscle simulations [7]. As represented in Fig. 10.2, the musculotendon model is generically described as a muscle actuator composed by a contractile element (CE) and a passive element (PE) assembled in series with a spring element of stiffness K^T , representing the tendon. The muscle actuator is angled by a pennation angle (α) to accommodate oblique muscle insertions. The pennation angle describes the angular orientation that muscle fibers present with respect to the aponeurosis, the internal portion of the tendon (Fig. 10.2a). The model assumes that all fibers that constitute a given muscle are parallel, straight, coplanar and present the same length (l^M). Additionally, it is also considered that the musculotendon element does not present mass or friction [8, 10].

Computationally, musculotendon lines of action (or lines of pull) are modeled as extensible lines with origin and insertion points connected to different elements representing the skeletal bones. Lines of action mediate the origin and insertion points

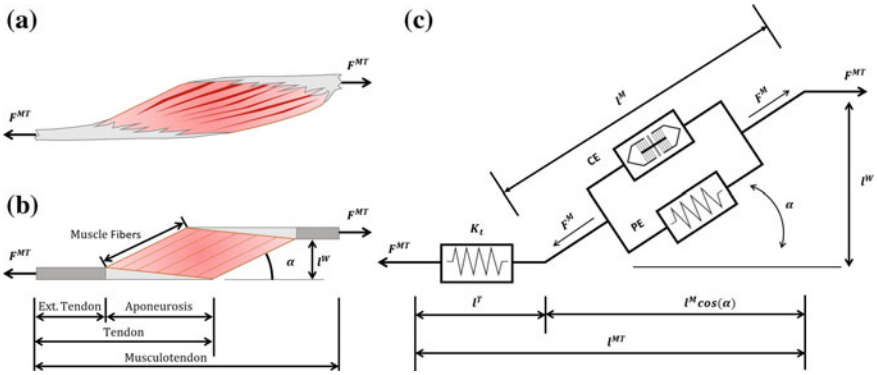


Fig. 10.2 The musculotendon model: **a** anatomical illustration of a unipennate muscle (muscle inserts obliquely into the tendon); **b** schematic representation of a unipennate musculotendon unit (muscle fibers considered as straight and parallel lines with equal length and orientation, attached to the tendon at the oblique aponeurosis region); **c** detailed representation of musculotendon unit considering the Hill-type element and the elastic tendon element

of the muscle actuator and can be defined as a collection of straight line segments angling at specific via points [20] or utilizing dedicated algorithms, which allow muscles to wrap around different types of geometrical obstacles representative of body segments or other biological structures [21]. The length of these lines, designated as musculotendon length (l^{MT}), varies in time according to the kinematics of the movement under analysis and its calculation is performed by the multibody dynamics formulation, for each time step, considering the respective biomechanical model under analysis. Geometrically, and taking in consideration Fig. 10.2c, the musculotendon length is defined as the sum of the tendon length (l^T) and the muscle fibers length (l^M), taking into account the pennation angle as represented in Eq. (10.1).

$$l^{MT} = l^T + l^M \cos(\alpha) \quad (10.1)$$

The muscle contractile element (CE) is assembled in parallel with a passive element (PE). The CE produces a force (F_{CE}^M) that depends on the peak isometric active force that the muscle can produce (F_0^M), on its activation level (a^M) that ranges from 0 (fully relaxed muscle) to 1 (fully activated muscle), and on the values of the force-length ($f_l(l^M)$) and force-velocity ($f_v(v^M)$) relationships (see Fig. 10.3a, b), which are dimensionless functions that describe the ability of the muscle to produce active contractile force taking in consideration its state (length and contraction velocity). The PE is included to represent the passive elastic properties of the muscle tissue, generating a force (F_{PE}^M) that depends solely on the muscle elongation.

Accordingly, the force developed by a muscle actuator (F^M) can be defined as the sum of the forces generated by these two elements [15, 22]:

$$F^M = F_{CE}^M + F_{PE}^M = f_l(l^M) f_v(v^M) F_0^M a^M + F_{PE}^M(l^M) \quad (10.2)$$

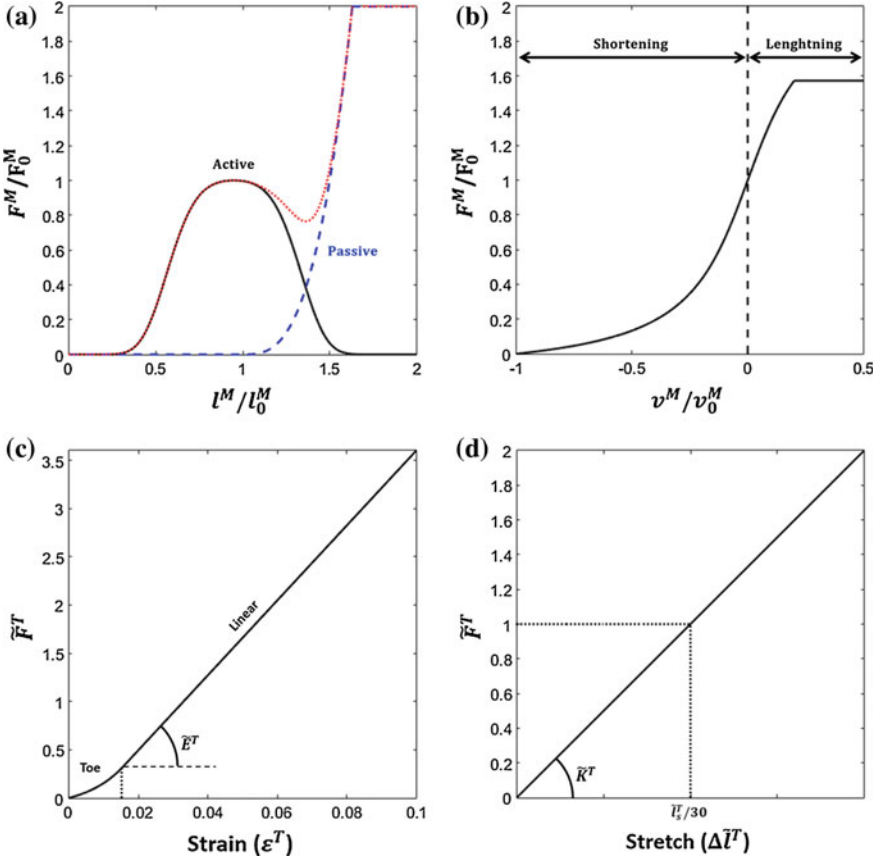


Fig. 10.3 Dimensionless muscle and tendon force properties: **a** muscle active and passive force-length relationship (fully activated muscle); **b** muscle force-velocity relationship (fully activated muscle); **c** tendon force-strain curve (adapted from [8]); **d** Tendon force-stretch curve (adapted from [8])

where the dimensionless force-length and force-velocity relationships are defined as:

$$f_l(l^M) = e^{\left[-\left[\frac{9}{4} \left(\frac{l^M}{l_0^M} - \frac{19}{20} \right) \right]^4 - \frac{1}{4} \left[\frac{9}{4} \left(\frac{l^M}{l_0^M} - \frac{19}{20} \right) \right]^2 \right]} \quad (10.3)$$

$$f_v(v^M) = \begin{cases} 0 & v^M < -v_0^M \\ -\frac{1}{\tan^{-1}(5)} \tan^{-1} \left(-5 \frac{v^M}{v_0^M} \right) + 1 & -v_0^M \leq v^M \leq 0.2v_0^M \\ \frac{\pi}{4 \tan^{-1}(5)} + 1 & v^M > 0.2v_0^M \end{cases} \quad (10.4)$$

and the passive element force is defined as:

$$F_{PE}^M(l^M) = \begin{cases} 0 & l^M < l_0^M \\ 8 \frac{F_0^M}{l_0^M} (l^M - l_0^M)^3 & l_0^M \leq l^M \leq 1.63l_0^M \\ 2F_0^M & l^M > 1.63l_0^M \end{cases} \quad (10.5)$$

It should be noted that the expressions presented in Eqs. (10.3)–(10.5) are not unique as other similar or equivalent ones can be found in the literature that perform the same function and provide similar results [7, 23, 24].

An important aspect of this muscle model, is that the calculation of the muscle force (F^M), for a specific muscle, depends solely on the values of four main muscle-specific parameters (optimal muscle fiber length (l_0^M), muscle maximum shortening velocity (V_0^M), optimal muscle fiber pennation angle (α_0) and peak isometric muscle active force (F_0^M)) and on the dimensionless relationships for force-length and force-velocity that describe the state-dependent nature of muscle force production.

Regarding the tendon, as this element is placed in series with the muscle element, the force developed by the musculotendon actuator is influenced by the dynamic equilibrium that is established between both components. The elongation of the tendon will affect directly the length of the muscle fibers, resulting in different values of force produced by the two force elements of the muscle actuator (CE and PE).

The influence of the tendon on the equilibrium depends on its mechanical properties, which encompass structural and material properties. When the stress-strain relationship ($\sigma^T(\varepsilon^T)$) is analyzed for the tendon, three distinct regions are observed in the curve (see Fig. 10.3c): the toe region, where for low strains ($\varepsilon^T \lesssim 2\%$) the value of the tendon tangent modulus of elasticity ($E^T = d\sigma^T/d\varepsilon^T$) increases non-linearly with the strain; the linear region, where the stress increases linearly with the strain (and E^T presents a constant value of approximately 1.2 GPa); and the failure region, where for values of strain near 10% the tendon enters in rupture [8]. It should be noted that the tendon strain (ε^T) is defined as the tendon stretch (Δl^T) per unit of tendon slack length (l_s^T), i.e., $\varepsilon^T = \Delta l^T / l_s^T$. The tendon slack length (l_s^T) is the length above which the tendon begins to produce force and the tendon stretch is defined as $\Delta l^T = (l^T - l_s^T)$.

The tendon element is commonly defined as a spring-like component with a constant stiffness K^T that depends on the physiological elastic properties of the tendon and to which is associated a relationship between tendon stress and stretch. A relevant aspect that should be highlighted in the present implementation is the dimensionless characteristic assumed to exist between the tendon force and the tendon strain and stretch, represented in Fig. 10.3c, d respectively. The use of such approach implies two assumptions: the first is that the elastic properties of the tendon are based on generic tendon material properties (taken from the stress-strain curve); the second is that when the tendon force is equal to the peak isometric active force of the muscle ($F^T = F_0^M$), which occurs for tendon strains (ε_0^T) near 0.033, the tendon specific stress (σ_0^T) is equal to 32 MPa, independently of the tendon being modelled, i.e., the tendon specific stress is musculotendon-independent [8].

Considering the normalization of the tendon stress by the tendon specific stress ($\tilde{\sigma}^T = \sigma^T / \sigma_0^T$), it is possible to define a generic relationship between $\tilde{\sigma}^T$ and ε^T that, with the exception of the tendon slack length (l_s^T), is independent of specific tendon characteristics. This relation enables also to obtain directly the normalized tendon force ($\tilde{F}^T = F^T / F_0^M$), as its value is equal to $\tilde{\sigma}^T$ [8]:

$$\tilde{\sigma}^T = \frac{\sigma^T}{\sigma_0^T} = \frac{F^T / A^T}{F_0^M / A^T} = \tilde{F}^T \quad (10.6)$$

A dimensionless relationship between force and strain in the tendon, as represented in Fig. 10.3c, can also be obtained from Eq. (10.6). Modelling the toe working region as an exponential function, it becomes possible to derive a generic expression that reflects this relation [9, 23]:

$$\tilde{F}^T = \begin{cases} \frac{\tilde{F}_{toe}^T}{e^{\tilde{K}_{toe}^T} - 1} \left(e^{\tilde{K}_{toe}^T \varepsilon_{toe}^T} - 1 \right) & \varepsilon^T \leq \varepsilon_{toe}^T \\ \tilde{F}_{toe}^T + \tilde{K}_{lin}^T (\varepsilon^T - \varepsilon_{toe}^T) & \varepsilon^T > \varepsilon_{toe}^T \end{cases} \quad (10.7)$$

with:

$$\tilde{K}_{lin}^T = \frac{1 - \tilde{F}_{toe}^T}{\varepsilon_0^T - \varepsilon_{toe}^T} \quad (10.8)$$

where \tilde{F}_{toe}^T and ε_{toe}^T are respectively the values of force and strain above which it is assumed that the tendon presents a linear behavior. The values for these two parameters vary in literature, ranging from 0.015 to 0.02 for ε_{toe}^T and 0.30 to 0.33 for \tilde{F}_{toe}^T [8, 9, 23]. Regarding \tilde{K}_{toe}^T , it represents a shape factor that controls the curvature of the exponential function, while \tilde{K}_{lin}^T adjusts the slope of the linear region, computed assuming the condition $\tilde{F}^T = 1$ when $\varepsilon_{toe}^T = 0.033$. Defining values for \tilde{F}_{toe}^T and \tilde{K}_{toe}^T , the remaining parameters of Eq. (10.7) can be calculated considering a continuity condition between the two branches of the equation. For the current work, the values presented in [9] ($\tilde{K}_{toe}^T = 1.3796$ and $\tilde{F}_{toe}^T = 0.3086$) are considered, yielding values of 0.0152 for ε_{toe}^T and 38.8427 for \tilde{K}_{lin}^T .

A disadvantage of this modelling procedure is the poor estimation of tendon stress for low values of strain, resulting in a wide range of values for (ε_0^T). This fact is a direct consequence of the high compliancy that tendons present for low levels of stress. Zajac considers that since the energy stored in these situations is very small, these differences are assumed to be embraced in the estimation of l_s^T [8].

Regarding the total force produced by the musculotendon actuator, due to the fact that muscle and tendon are assembled in series (as depicted in Fig. 10.2c), the resultant musculotendon actuator force is equal to the force transmitted by the tendon, which is, by its turn, equal to the force produced by the muscle, projected onto the line of action of the tendon:

$$F^{MT} = F^T = F^M \cos(\alpha) \quad (10.9)$$

The dynamic behavior of the musculotendon model can be described by a first order differential equation that relates the variation of the tendon force (F^T) with the tendon velocity (v^T) and stiffness (K^T) [8, 9]:

$$\frac{dF^T}{dt} = K^T \frac{dl^T}{dt} = K^T v^T \quad (10.10)$$

The normalized tendon force (\tilde{F}^T) is obtained rewritten Eq. (10.10) in its dimensionless form:

$$\frac{d\tilde{F}^T}{dt} = \tilde{K}^T \tilde{v}^T \quad (10.11)$$

with:

$$\begin{aligned} \tilde{K}^T &= K^T \frac{1/F_0^M}{1/l_0^M} \cong \frac{30}{\tilde{l}_s^T} \\ \tilde{v}^T &= v^T \frac{1}{v_0^M} \end{aligned} \quad (10.12)$$

where \tilde{K}^T and \tilde{v}^T represent respectively the dimensionless tendon stiffness and the normalized tendon velocity. From the analysis of the dimensionless tendon force-stretch curve depicted in Fig. 10.3d, the value of \tilde{K}^T can be estimated considering the slope of the respective curve ($\sim 30/\tilde{l}_s^T$). This relation assumes that for all working regions, the normalized tendon force varies linearly with the tendon stretch [8]. The tendon velocity (v^T) can be defined in terms of the musculotendon contraction velocity (v^{MT}) and muscle contraction velocity (v^M), through the differentiation of Eq. (10.1) with respect to time:

$$v^{MT} = v^T + v^M \cos \alpha - l^M \dot{\alpha} \sin \alpha \quad (10.13)$$

Previous equation presents a dependency on α and $\dot{\alpha}$ (the pennation angular velocity). For the calculation of the latter, the constant isovolume assumption is considered [8]:

$$l^W = l_0^M \sin \alpha_0 = l^M \sin \alpha \quad (10.14)$$

This approximation considers that a constant relation exists between muscle length and pennation angle and that such relation is equal to the muscle width (l^W), which can be calculated from the optimal muscle fiber length (l_0^M) and optimal pennation angle (α_0). For this assumption to be valid, the musculotendon model should consider that aponeurosis thickness increases from the internal to external portion, as represented in Fig. 10.2a, b, such that all fibers are parallel and equal in length [8, 10]. The derivative of Eq. (10.14) with respect to time yields in an expression for the pennation angular velocity:

$$\dot{\alpha} = -\frac{v^M \sin \alpha}{l^M \cos \alpha} \quad (10.15)$$

which can be substituted in Eq.(10.13) to obtain a simplified expression for the tendon actuator velocity:

$$v^T = v^{MT} - \frac{v^M}{\cos(\alpha)} \quad (10.16)$$

From the analysis of previous equation, it can be seen that the calculation of the tendon velocity (v^T), which is needed for the calculation of the derivative of the tendon force as expressed in Eq.(10.11), requires the previous calculation of v^{MT} , v^M and α . While the musculotendon contraction velocity (v^{MT}) can be retrieved from the kinematics of the system, which are managed at the level of the multibody dynamics solver, the values of the muscle velocity (v^M) and pennation angle (α) are themselves influenced by the state of the tendon, fact that brings an additional level of complexity to the problem. The procedure adopted in this work to calculate these variables is included on the musculotendon model used to integrate the differential equation describing the tendon dynamics (Eq. (10.10)). This model is schematically represented in Fig. 10.4 and the necessary steps for its implementation described hereafter.

Considering that Eq. (10.11) is integrated as an initial value problem (IVP) using a direct integration (DI) method, the musculotendon model requires as input, for the present time step, the values of the normalized tendon force (\tilde{F}_a^T) for a fully activated state of the muscle (i.e., with $a^M = 1$), and the length (l^{MT}) and contraction velocity (v^{MT}) of the musculotendon actuator. The purpose now is to calculate, also for the present time step, the derivative of the normalized tendon force for a fully activated state of the muscle ($\dot{\tilde{F}}_a^T$) that once integrated in time provides the value of the tendon normalized force (\tilde{F}_a^T) for the next time step. It should be noted that this integration step is required independently of the type of dynamic analysis being performed (i.e., forward or inverse). Regarding the estimation of the initial values for normalized tendon force (\tilde{F}_a^T), this can take advantage of the underlying multibody solver and, hence, discussed in the next section.

In order to enable the calculation of the muscle velocity and pennation angle, the tendon length and the contractile and passive muscle forces must be firstly determined. The first step comprehends the determination of the tendon length taking into account its own elastic properties (as depicted in Fig. 10.3c, d). The normalization of tendon slack length by the optimal fiber length, denoted by \tilde{l}_s^T , defines the compliance of the tendon. The influence of the tendon in the musculotendon actuator can be predicted from the value of \tilde{l}_s^T . Low values of \tilde{l}_s^T denote stiff (rigid) tendons ($\tilde{l}_s^T \leq 1$), while higher values indicate compliant tendons ($\tilde{l}_s^T > 1$) [8]. Rigid tendons are treated as inextensible elements with lengths that do not change over time and, as so, their velocity is set to zero. On the other hand, the length of a compliant tendon varies according the normalized musculotendon force that acts on the system. Its value can be computed using the inverse function of the force-strain curve, depicted in Fig. 10.3c and presented in Eq.(10.7) [8, 9]:

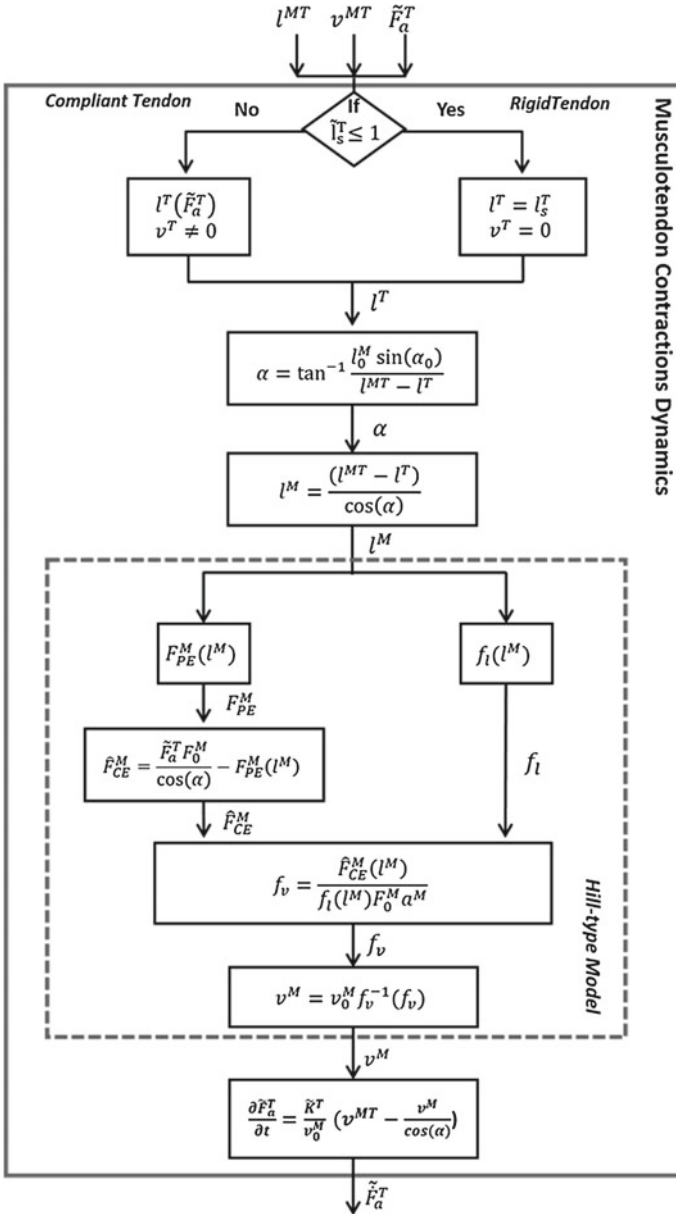


Fig. 10.4 Flowchart describing the musculotendon model: methodology used to compute the derivative of the normalized tendon force and remaining model parameters for current time step

$$l^T(\tilde{F}^T) = \begin{cases} l_s^T \left(1 + \frac{\varepsilon_{toe}^T}{\tilde{K}_{toe}^T} \ln \left(\frac{e^{\tilde{K}_{toe}^T} - 1}{\tilde{F}_{toe}^T} \tilde{F}^T + 1 \right) \right), & 0 \leq \tilde{F}^T \leq \tilde{F}_{toe}^T \\ l_s^T \left(1 + \frac{\tilde{F}^T - \tilde{F}_{toe}^T}{\tilde{K}_{lin}^T} + \varepsilon_{toe}^T \right), & \tilde{F}^T > \tilde{F}_{toe}^T \end{cases} \quad (10.17)$$

The second step comprises the estimation of the pennation angle by applying the muscle constant isovolume assumption described by Eq. (10.14). Substituting Eq. (10.1) in Eq. (10.14) yields an expression dependent of l^T and l^{MT} :

$$\alpha = \tan^{-1} \frac{l^w}{l^{MT} - l^T} \quad (10.18)$$

The muscle length (l^M) can now be calculated by means of Eq. (10.1) and, with its value, the calculation of $f_l(l^M)$ and $F_{PE}^M(l^M)$, as indicated in Eqs. (10.3) and (10.5), can also be carried out.

The next step comprehends the calculation of the muscle contraction velocity (v^M). First, Eq. (10.2) is solved for the force-velocity relationship $f_v(v^M)$. Then, using Eq. (10.9), the force of the contractile element can be expressed as a function of \tilde{F}^T and of the passive element force (F_{PE}^M) calculated previously, resulting:

$$f_v(v^M) = \frac{F_{CE}^M}{f_l(l^M)F_0^M a^M} = \left(\frac{\tilde{F}^T}{\cos(\alpha)} F_0^M - F_{PE}^M(l^M) \right) \frac{1}{f_l(l^M)F_0^M a^M} \quad (10.19)$$

A fact that steams directly from the observation of Eq. (10.19) is the possibility of appearing three numerical singularities when $\alpha \rightarrow \pi/2$, $a^M \rightarrow 0$ and $f_l \rightarrow 0$. A simple approach to mitigate this problem considers the definition of boundaries for these variables, so that these singularities are never reached, i.e., $\alpha < \pi/2$, $a^M > 0$ and $f_l > 0$. Millard et al. reported that the use of these conditions can result in a numerical stiff problem, particularly when these limits are approached, increasing substantially the time required in the integration step [10]. In order to avoid the appearance of these singularities and recalling that the aim of the present procedure is the calculation of the derivative of the normalized tendon force for a fully activated state of the muscle (\tilde{F}_a^T), maximum activation ($a^M = 1$) is assumed for the calculation of $f_v(v^M)$, and limits are set for the maximum pennation angle ($\alpha < \cos^{-1}(0.1)$) and for the minimum muscle force-length relationship ($f_l > 0.1$) [10].

Muscle contraction velocity is now calculated inverting the force-velocity relationship curve. For computational reasons, Eq. (10.4) is best-fitted by a sigmoid curve and solved with respect to the muscle contraction velocity, yielding:

$$v^M(f_v) = v_0^M \left(0.15 \log \left(\frac{f_v}{-f_v + 1.572} \right) - 0.08 \right) \quad (10.20)$$

The final step comprehends the calculation of the derivative of the normalized tendon force for a fully-activated state (\tilde{F}_a^T), combining Eqs. (10.11), (10.12) and (10.16):

$$\tilde{F}_a^T = \frac{\tilde{K}^T}{v_0^M} \left(v^{MT} - \frac{v^M}{\cos \alpha} \right) \quad (10.21)$$

The value of \tilde{F}_a^T is now ready to be numerically integrated, allowing to estimate \tilde{F}_a^T for the next time step. This integration step can be done simultaneously with the integration step of the multibody dynamics solver in a process that is described in the following section.

10.2.2 Incorporation of the Musculotendon Model in a Multibody Dynamics Framework

The incorporation of the musculotendon model in a general multibody formulation is carried out considering musculotendon forces as external forces, which are included in the equations of motion of the system (EoM) as equivalent generalized forces (\mathbf{g}^{MT}), applied over the generalized coordinates describing the biomechanical model [15, 25]:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{g}^{ext} + \mathbf{g}^{MT} = \mathbf{g}^{ext} + \sum_{i=1}^{nm} \hat{\mathbf{g}}_{CE}^{MTi} a^{M_i} \cos(\alpha_i) + \sum_{i=1}^{nm} \mathbf{g}_{PE}^{MTi} \cos(\alpha_i) \quad (10.22)$$

where \mathbf{M} , $\ddot{\mathbf{q}}$, $\Phi_{\mathbf{q}}^T$ and $\boldsymbol{\lambda}$ represent the global mass matrix, the vector of generalized accelerations, the Jacobian matrix of the kinematic constraints and the vector of Lagrange multipliers respectively. Vectors $\hat{\mathbf{g}}_{CE}^{MTi}$ and \mathbf{g}_{PE}^{MTi} represent, for musculotendon i , the generalized contractile muscle force vector for a fully-activated muscle state and the generalized passive muscle force vector respectively. Vector \mathbf{g}^{ext} represent all other generalized external forces applied over the biomechanical system.

When the musculotendon model is utilized in a forward dynamics simulation, the model allows for the calculation of the forces generated in the different elements of the Hill-type muscle model and tendon as well as their actual state (lengths and contraction velocities), given as input the state of the musculotendon element (l^{MT} and v^{MT}), the prescribed muscle activations (a^M) and the normalized musculotendon force for a fully activated muscle state (\tilde{F}_a^T) for the current time step. The musculotendon force is applied as an external force in the EoM of the system and \tilde{F}_a^T is calculated for each actuator. Then \tilde{F}_a^T is fed into the integrator, along with the velocities ($\dot{\mathbf{q}}$) and accelerations ($\ddot{\mathbf{q}}$) of the system in the current time step, which are then integrated together to yield the state of the system and musculotendon model for the next time step, as represented in Fig. 10.5a.

When applied in an inverse dynamics analysis (see Fig. 10.5b), the model allows for the calculation of the forces produced by the contractile and passive elements, considering as input the system kinematics and \tilde{F}_a^T computed also for the current

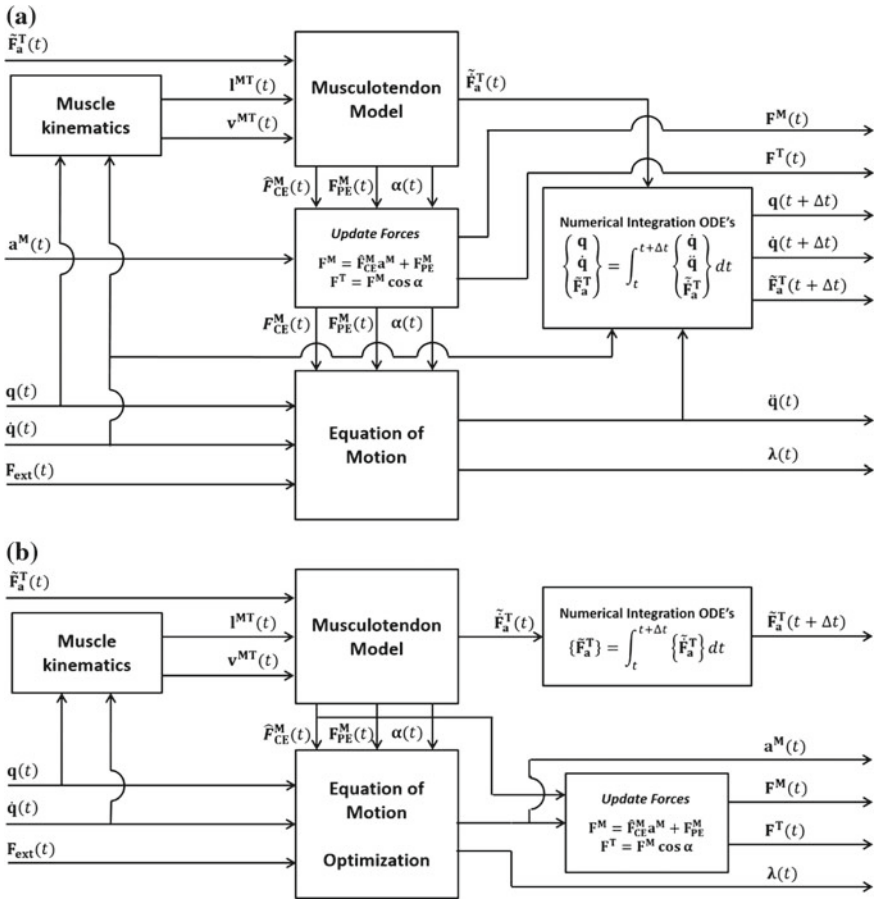


Fig. 10.5 Incorporation of the musculotendon model in a multibody dynamics formulation: **a** forward-dynamics simulation; **b** inverse-dynamics analysis

time step. From the input variables the musculotendon model calculates the state of all the elements of the musculotendon actuator as well as the value of \tilde{F}_a^T that is numerically integrated to obtain the value of \tilde{F}_a^T for the next instant. The forces produced by the musculotendon actuators will be applied in the EoM as external forces.

As the muscle activations are unknown, the system dynamics and the muscle-related variables are estimated by means of a static optimization step that, for the current time, calculates the optimal set of activations that minimizes a specified physiological based criterion. The activations calculated for the previous time step are used as an initial approximation for the optimization procedure.

The optimization problem for the calculation of the muscle activations is stated as:

$$\begin{aligned}
 &\text{Find : } \bar{\lambda} = \begin{Bmatrix} \lambda \\ \mathbf{a}^M \end{Bmatrix} \\
 &\text{Minimize : } \mathfrak{S}(\bar{\lambda}) \\
 &\text{Subject to : } \begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda - (\mathbf{g}^{ext} + \mathbf{g}^{MT}) = \mathbf{0} \\ \lambda^{lb} \leq \lambda \leq \lambda^{ub} \\ \mathbf{0} \leq \mathbf{a}^M \leq \mathbf{1} \end{cases}
 \end{aligned} \tag{10.23}$$

where $\bar{\lambda}$ is the optimization vector composed by the vector of state variables (λ) and by the vector of control variables (\mathbf{a}^M), and $\mathfrak{S}(\bar{\lambda})$ is the objective function expressing the physiological criterion to minimize (e.g. metabolic energy, muscle stress, muscle fatigue, among others). The minimization of the objective function is carried out subjected to the fulfillment of the set of in/equality optimization constraints that include the EoM of the system, written in the homogeneous form, and lower and upper bounds for the state and control variables. A detailed description of the applied optimization methodology can be consulted in [11, 15].

Once the activations are determined, the muscle and tendon forces as well as muscle and tendon states can be calculated and updated for the current time step and the analysis may proceed for the next time step. It should be noted that with these updated values, a new optimization step can be carried out to obtain new optimized activations, in an iterative process that comes to an end when the residual of activations, between previous and actual iteration, is within a user specified value. Although it is expected that this process would improve the biomechanical relevance of the results, it also significantly increases the computational effort and execution computational times, due to the extra optimization steps, and, therefore, was not implemented in the present work.

For both types of dynamic analyses (forward and inverse), an initial estimation of \tilde{F}_a^T is required to compute the initial tendon length (l^T) or vice-versa. Since both these values are usually absent in the first time step, the adopted procedure to obtain this initial estimation was to consider rigid tendons with lengths equal to the slack length and consequently with null tendon velocities. This allows the initial estimation of the musculotendon state and, consequently, the initial estimation of \tilde{F}_a^T . An iterative optimization procedure, similar to the one referred previously, can be considered that corrects the initial estimation of \tilde{F}_a^T by feeding back its value into the musculotendon model but this time considering compliant tendons. This correction procedure was not implemented in this work.

10.3 Results and Discussion

The methodology presented before was incorporated in APOLLO, a three-dimensional multibody dynamics software, described with natural coordinates, developed in-house [15]. The human articulated biomechanical model, represented in

Fig. 10.6 Biomechanical model (lower limbs + HAT) applied in the inverse dynamic analysis of the three selected movements: **a** 3D representation of the kinematic structure of the biomechanical model; **b** representation of the 86 musculotendon actuators used in the lower limbs muscle apparatus

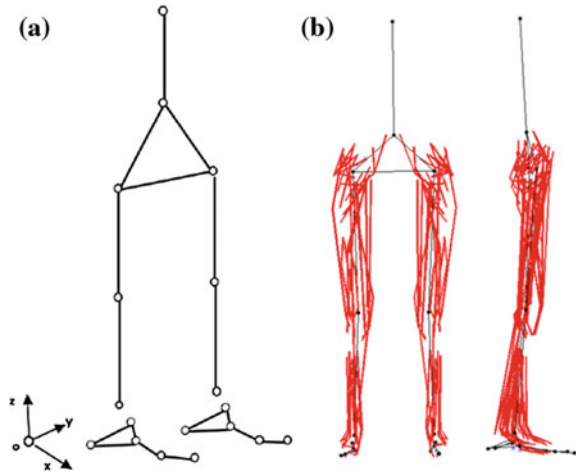


Fig. 10.6a, composed by 12 segments (2 three-segment feet, 2 shanks, 2 thighs, 1 pelvis and 1 HAT) was constructed. The model makes use of 20 rigid bodies, defined with 25 points and 29 unit vectors, and allows a total of 27 degrees-of-freedom. In order to represent in detail the muscle apparatus of the lower limbs, 86 musculotendon actuators (43 per leg) were integrated, as represented in Fig. 10.6b, considering the muscle and tendon parameters presented in [26].

The behavior of the presented musculotendon model was evaluated by applying it to the analysis of the musculotendon dynamics of the biomechanical model for three different activities: walking, running and jumping. The choice of movements had in consideration the different levels of tendon recruitment that are expected for these activities. The acquisition of the experimental data for the three movements was performed in the *Laboratório de Biomecânica de Lisboa* (Lisbon Biomechanics Laboratory) at Instituto Superior Técnico. A male subject (23 years old, 76 kg, 1.78 m) with no history of gait disorders was recruited among the university students to perform the selected movements. The subject was informed of the objectives of the study and informed consent was obtained. The acquisition of kinematic data made use of a three-dimensional motion capture system, composed by 14 infrared reflective cameras (Qualisys ProReflex MC1000). The sagittal view of the three reconstructed movements is represented in Fig. 10.7.

Three force plates (AMTI OR 6-7-1000, 508 mm × 464 mm) were used to acquire the ground reaction forces (GRF). Markers kinematics and GRF were acquired synchronously using Qualisys Track Manager 2.9 software. The sampling frequency was set to 100 Hz for the IR cameras and 1000 Hz for the force plates. The marker set protocol was designed to allow the acquisition of the six DOFs of each rigid body, considering for that 56 passive markers placed in the anatomical landmarks suggested in [27]. A detailed description of the protocol and biomechanical model can be found in [11]. A set of 10 valid trials were acquired for each activity. The statistical treatment was performed using MATLAB software.

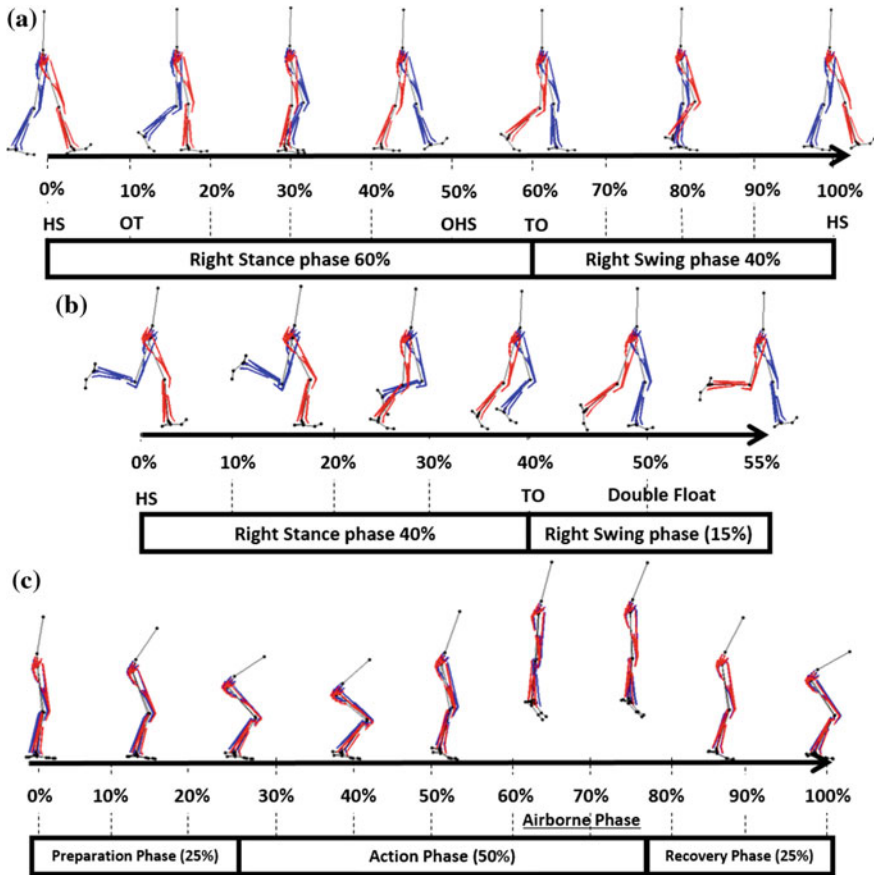


Fig. 10.7 Movements kinematics and respective phases: **a** walking **b** running **c** jumping

In order to study the influence of the tendon in the musculotendon model, the output of the analysis for the three activities is compared with the results obtained with a model in which the tendons are considered infinitely stiff, i.e. with constant length equal to the slack length. Due to the large amount of data, the results presented in this work will focus only on the principal muscles that constitute the antagonist muscle groups responsible for controlling ankle plantar flexion (gastrocnemius lateral (GL) and medial (GM) and soleus) and ankle dorsiflexion (tibialis anterior (TA), extensor digitorum longus (EDL) and extensor hallucis longus (EHL)). The results obtained for the remaining muscle apparatus can be consulted in [11].

In general, patterns of force and activation, generated by the model with compliant tendons, are consistent with the performed movements. The analysis of the gait results (Fig. 10.8a, b) for the triceps surae group allows the observation of a peak of force and activation during the mid-stance (~ 30 – 50 % of gait cycle (GC)) and pre-swing (~ 50 – 60 % of the GC) phases. It is known that the mid-stance is charac-

terized by a storage of elastic energy [28], which can be identified by observing the increase of musculotendon force in these muscles. The stored energy is posteriorly released during the push-off phase (40–60 % of the GC). This phase is also characterized by a concentric contraction of these muscles that plantar flexes the ankle and propels the body forward. This sequence of events is also observed in the force and activation curves for this phase. The results also show an increase of force and activation values until approximately 40–50 % of GC, followed by a decrease during the last moments of the propulsion phase. Moreover, similar curves are obtained when the activation patterns are compared with electromyographic (EMG) data [28]. For instance, the analysis of the EMG signal for the triceps surae muscles shows also a peak between the 30–50 % of GC, in line with the powered plantar flexion that occurs during propulsion. A consistent pattern of recruitment is also observed in the running and jumping movements (as depicted in Fig. 10.8c–f). The phases characterized by energy storage and concentric contraction present also the expected variations in the musculotendon force and activation (running: 10–40 %, jumping: 20–45 %). A consistent peak of force and activation is also observed in the jumping movement during the recovery phase (see Fig. 10.8e, f for 80–100 % of cycle). This phase is characterized by an eccentric contraction of the triceps surae muscles to absorb the energy of the impact and control the dorsiflexion motion that occurs until both feet are flat.

The force and activation curves for the dorsiflexors group present also consistent patterns with the performed motions. The analysis of the tibialis anterior dynamics during the gait (Fig. 10.9a, b), allows to observe a peak of force and activation in the first instants of the cycle (0–10 % of GC), while the ankle is plantar flexing. This eccentric contraction allows to counteract the effect of the resultant ground reaction forces, which are in this phase located in a posterior position to the ankle joint, controlling it and avoiding an abrupt plantar flexion (foot slap) usually observed in subjects suffering from weakness of the foot and ankle dorsiflexors [28, 29]. This effect can also be observed in the force and activation curves obtained for running (Fig. 10.9c). During approximately the first 25 % of the cycle, an activation of the TA is observed in order to assist the controlled plantarflexion of the ankle after the contact of the foot with ground. Other expected peaks are also observed in the TA activation curve during the swing phase of the gait cycle (Fig. 10.9a, b). In this event, the TA is active to support the foot in the air ensuring a safe toe clearance and a dorsiflexed foot that guarantees an initial contact made with the heel. The analysis of the activation patterns for jumping movement (Fig. 10.9e) allows also to observe an action of dorsiflexor muscles, essentially to ensure body equilibrium and ankle joint control during propulsion generation (10–60 %).

Comparative analysis of the results produced by the compliant tendon model with those obtained by the rigid tendon model shows significant differences on the pattern and magnitude of the force and activation curves. The model with rigid tendons presents non-physiological muscle forces caused by the passive component of the Hill-type model more frequently than its counterpart (e.g. Figs. 10.8b, d, e and 10.9d). On the other hand, the comparison between the forces produced by the contractile elements and the musculotendon forces for the compliant model (Figs. 10.8a, c, e and

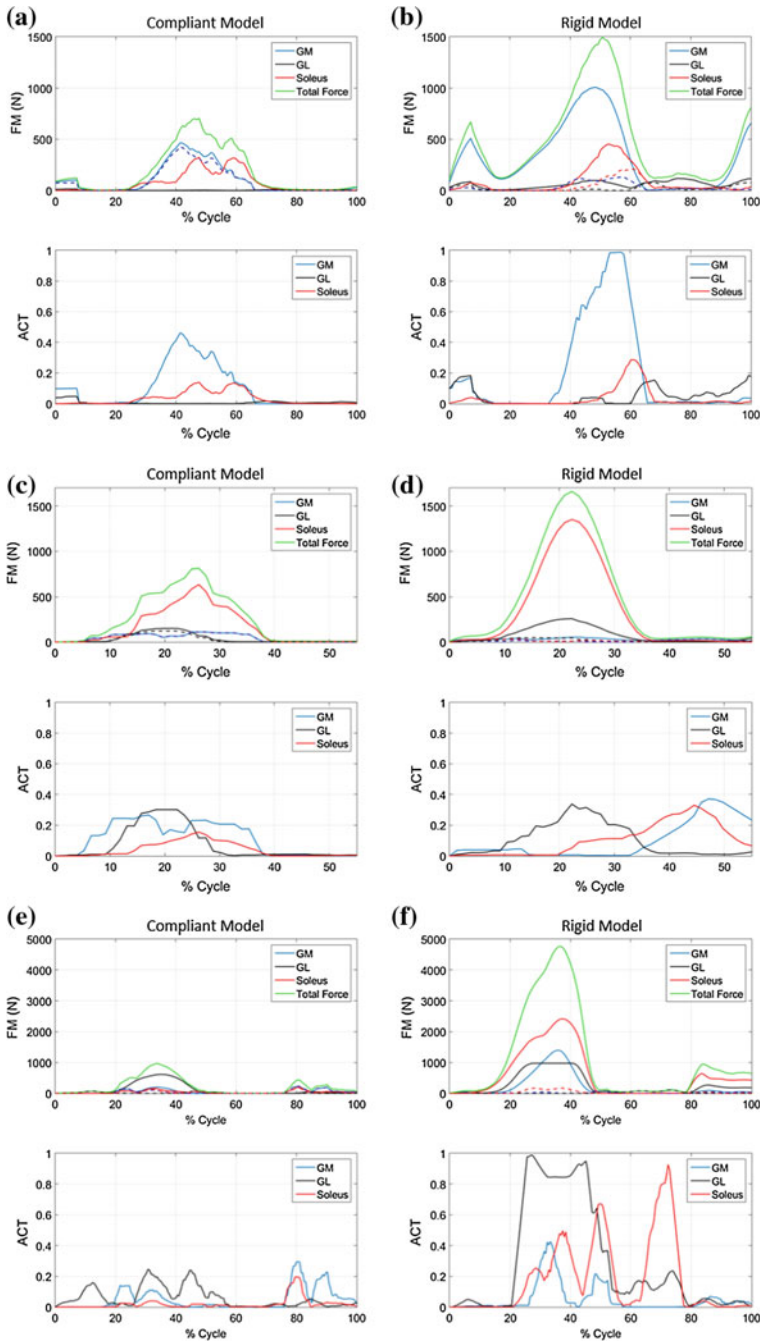


Fig. 10.8 Muscle force (*solid line*), contractile muscle force (*dashed line*) and muscle activation of the triceps surae group obtained using the compliant (*left*) and rigid (*right*) models: **a, b** walking; **c, d** running and **e, f** jumping

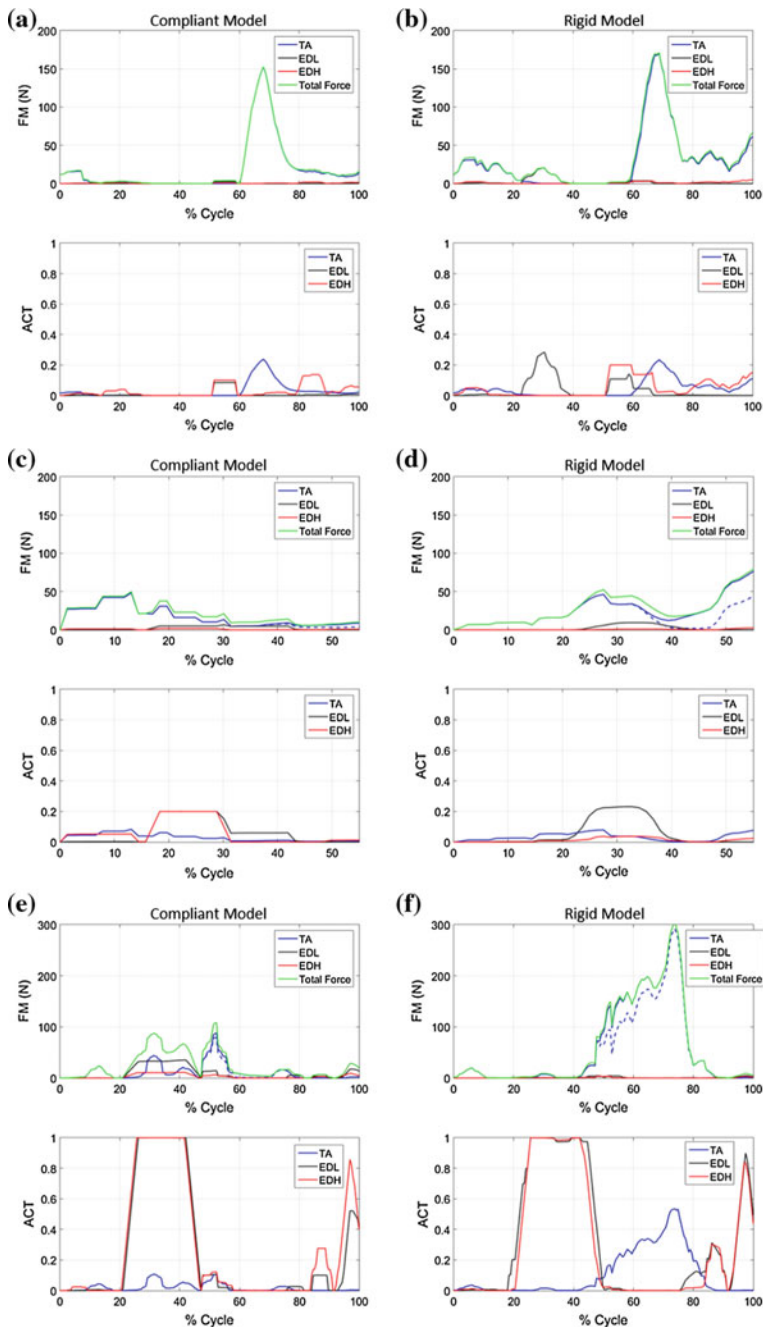


Fig. 10.9 Muscle force (*solid line*), contractile muscle force (*dashed line*) and muscle activation of the dorsiflexor muscles group obtained using the compliant (*left*) and rigid (*right*) musculotendon model: **a, b** walking; **c, d** running and **e, f** jumping

10.9a, c, e) shows small differences, indicating that passive forces are almost non-existent in this model. Moreover, despite presenting lower values of contractile forces, the activations for the rigid tendon model are higher. This fact is understandable since as the muscle is working away from its optimal length (l_0^M), it requires a higher activation level to achieve the same contractile force. For instance, the magnitudes of the contractile forces obtained for the gastrocnemius medialis for the gait example are respectively 500N for an activation of 0.4 (compliant model) and 135N for an activation of 1.0 (stiff model).

The results also show that the inclusion of a compliant tendon influences the contraction dynamics of the muscle by avoiding its excessive extension. This behavior was also expected since modelling the tendon as an elastic element prevents the muscle actuator to be the only one to mediate all the length variations experienced by the musculotendon element. Results indicate that the influence of modelling the tendon as an elastic element is as much important as the level of muscle recruitment and forces involved (ex: running and jumping).

10.4 Conclusions

A musculotendon model was presented that allows the determination of the forces produced by the several components of the model as well as their respective state. The process of incorporating the model in a multibody dynamics formulation was also explained both for inverse and forward dynamic analysis of multibody systems.

It is important to note that the inclusion of the tendon element in muscle models increases considerable the complexity of the problem to solve, as the force developed by the musculotendon element is influenced by the dynamic equilibrium that is established between the two elements. Instead of considering only algebraic equations to describe muscle contraction dynamics [15], tendon modelling requires the solution of a first order system that relates the force developed by this element and its velocity. An intricate step required for the solution of this problem was the calculation of the muscle actuator velocity, which is needed to evaluate the tendon velocity by means of the inverse force-velocity relationship. Besides requiring the activation state for that time step, the inverse function is also susceptible to the appearance of singularities, which can turn the system in a numerical stiff problem.

To avoid this problem, particularly in the inverse dynamic analysis in which the muscle activations are unknown, the methodology proposed in this work considers the system in a full-activated state. Independently of the type of analysis, the maximum available force in tendon, and consequently the maximum force that can be developed by the muscle actuator, is computed for each time step, being then updated by the prescribed activations if the model is run in a forward dynamics simulation or optimized if applied in an inverse dynamics perspective. For the computation of the maximum available force, the model requires the kinematics of the musculotendon actuator and the normalized tendon force for a fully activated state, which needs to be previously determined. The presented musculotendon model allows also to calculate

the derivative of the normalized tendon force for a fully activated state, which, when fed in the integrator, enables the calculation of the normalized tendon force for a fully activated state for the next time step.

The performance of the musculotendon model presented in this work was examined by applying it in the inverse dynamic analysis of three daily activities with different levels of tendon recruitment: walking, running and jumping. The model outputs were compared with a rigid musculotendon model. The force and activation curves were consistent with the performed movement and the respective EMG patterns. The comparison of the results obtained for the two musculotendon models, rigid and compliant, indicates a clear influence of the tendon in the musculotendon contraction dynamics for all the studied movements. In general, the inclusion of a compliant tendon allows muscle to work nearer its optimal zone, as variations on the musculotendon element are not supported solely by the muscle actuator, avoiding also non-physiological passive forces resultant of the excessive extension of the muscle component.

It is important to note that representing the tendon as a spring element in which the stiffness is only dependent of l_s^T and F_0^M presents some limitations. Despite allowing its application to large scale analysis, the musculotendon actuator is not tendon-specific. The material properties (ε_0^T and σ_0^T), which are used to define the dimensionless tendon force-stretch/strain relationship, were obtained from reasonable values from a range of experimental data. However, the proposed model can be adapted to a specific tendon through the definition of a force-stretch relationship dependent on its material properties [8]. Moreover, besides all the assumptions described in Sect. 10.2, the current model does not account for adaptations of the force-length and force-velocity curves or on the isometric active muscle force as the result of the muscle activation history [8, 30, 31]. Other conditions, such as temperature [32] or fatigue [33], which proved to influence also the muscle contraction dynamics, are not considered by the proposed methodology. However, it is possible to found in literature some works discussing methodologies to account for variations in these conditions (e.g. muscle history [31] and fatigue [34, 35]).

Static optimization methodologies were applied in the solution of the muscle redundancy problem. Although the adaption of the presented methodology to the use of dynamic optimization techniques will certainly increase the computational time, it will also improve the relevance of the results, as it will allow to consider activation dynamics and account for the electromechanical delay between the generation of the neural signal and the muscle contraction.

A final remark to refer that it is well known that musculotendons present a viscoelastic behavior [36]. To include that characteristic in the present model, a damping term needs to be included that will allow to consider some energy dissipation, which is known to occur in these systems.

In conclusion, despite increasing the complexity of the problem to solve, results showed that the inclusion of tendons in the calculation of muscle forces renders results with higher physiological relevance, suggesting that its use should not be neglected, even in the study of slower movements such as walking.

Acknowledgments The authors would like to thank Fundação para a Ciência e Tecnologia (FCT) for the support given through Ph.D. grant (SFRH/BD/51574/2011).

References

1. Hoy MG, Zajac FE, Gordon ME (1990) A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment arm on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle. *J Biomech* 23(2):157–169
2. Luzio de Melo P, da Silva MT, Martins J, Newman D (2015) A microcontroller platform for the rapid prototyping of functional electrical stimulation-based gait neuroprostheses. *Artif Organs* 39(5):E56–66
3. Magnusson SP, Narici MV, Maganaris CN, Kjaer M (2008) Human tendon behaviour and adaptation, in vivo. *J Physiol* 586(1):71–81
4. Thelen DG, Chumanov ES, Best TM, Swanson SC, Heiderscheit BC (2005) Simulation of biceps femoris musculotendon mechanics during the swing phase of sprinting. *Med Sci Sports Exerc* 37(11):1931–1938
5. Huxley AF (1957) Muscle structure and theories of contraction. *Prog Biophys Biophys Chem* 7:255–318
6. Hill AV (1938) The heat of shortening and the dynamic constants of muscle. *Proc R Soc Lond B Biol Sci* 126(843):136–195
7. Buchanan T, Lloyd D (2004) Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command. *J Appl Biomech* 20(4):367–395
8. Zajac FE (1989) Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit Rev Biomed Eng* 17(4):359–408
9. Martin CF, Schovanec L (1999) The control and mechanics of human movement systems. *Dynamical systems, control, coding, computer vision*. Springer, New York, pp 173–202
10. Millard M, Uchida T, Seth A, Delp SL (2013) Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J Biomech Eng* 135(2):21005-1-11
11. Oliveira AR (2014) Development of a musculotendon model within the framework of a multi-body systems dynamics, MSc thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa
12. Komura T, Prokopow P, Nagano A (2004) Evaluation of the influence of muscle deactivation on other muscles and joints during gait motion. *J Biomech* 37(4):425–436
13. Salinas-Avila A, Moreno-Valenzuela J (2009) Switched control of mechanical systems by using musculotendon actuators. In: 2009 American Control Conference. IEEE, pp 331–336
14. Delp SL, Loan JP (1995) A graphics-based software system to develop and analyze models of musculoskeletal structures. *Comput Biol Med* 25(1):21–34
15. Silva MT (2003) Human motion analysis using multibody dynamics and optimization tools. Instituto Superior Técnico - Universidade Técnica de Lisboa
16. Pereira AF (2009) Development of a hill-type model with fatigue for the calculation of the redundant muscle forces using multibody dynamics. MSc thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa
17. Hall JE, Guyton A (2010) *Textbook of medical physiology*. Elsevier Health Sciences
18. Pandy MG, Barr RE (2004) *Biomechanics of the musculoskeletal system*. Biomedical Engineering and Design Handbook
19. Romero F, Melo PL, Silva MT, Alonso FJ (2015) Validation of an artificially activated mechanistic muscle model by using inverse dynamics analysis. *Mech Mach Theory* 93:1–10
20. Ambrósio JA, Silva M (2005) A biomechanical multibody model with a detailed locomotion muscle apparatus. *Adv Comput Multibody Syst* 155–184

21. Garner BA, Pandy MG (2000) The obstacle-set method for representing muscle paths in musculoskeletal models. *Comput Methods Biomech Biomed Eng* 3(1):1–30
22. Kaplan ML (2000) Efficient optimal control of large-scale biomechanical systems. PhD thesis, Stanford University
23. Thelen DG (2003) Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *J Biomech Eng* 125(1):70–77
24. Winters JM (1990) Hill-based muscle models: a systems engineering perspective. Multiple muscle systems. Springer, New York
25. De Jalon JG, Bayo E (1994) Kinematic and dynamic simulation of multibody systems. Springer, New York
26. Yamaguchi GT (2005) Dynamic modeling of musculoskeletal motion: a vectorized approach for biomechanical analysis in three dimensions. Springer Science & Business Media
27. Wu G, Siegler S, Allard P, Kirtley C, Leardini A, Rosenbaum D, Whittle M, D’Lima DD, Cristofolini L, Witte H et al (2002) ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine. *J Biomech* 35(4):543–548
28. Winter DA (1991) Biomechanics and motor control of human gait: normal, elderly and pathological, University of Waterloo Press
29. Blaya JA, Herr H (2004) Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Trans Neural Syst Rehabil Eng* 12(1):24–31
30. Herzog W, Leonard T (2002) Force enhancement following stretching of skeletal muscle a new mechanism. *J Exp Biol* 205(9):1275–1283
31. Meijer K, Grootenboer HJ, Koopman HFJM, van der Linden BJJJ, Huijijng PA (1998) A Hill type model of rat medial gastrocnemius muscle that accounts for shortening history effects. *J Biomech* 31(6):555–563
32. Rall JA, Woledge RC (1990) Influence of temperature on mechanics and energetics of muscle contraction. *Am J Physiol* 259(2 Pt 2):R197–203
33. Stephens JA, Taylor A (1972) Fatigue of maintained voluntary muscle contraction in man. *J Physiol* 220(1):1–18
34. Xia T, Frey Law LA (2008) A theoretical approach for modeling peripheral muscle fatigue and recovery. *J Biomech* 41(14):3046–3052
35. Pereira AF, Silva MT, Martins JM, de Carvalho M (2011) Implementation of an efficient muscle fatigue model in the framework of multibody systems dynamics for analysis of human movements. *Proc Inst Mech Eng Part K J Multi-body Dyn* 225(4):359–370
36. Taylor DC, Dalton JD, Seaber AV, Garrett WE (1990) Viscoelastic properties of muscle-tendon units: The biomechanical effects of stretching. *Am J Sports Med* 18(3):300–309

Chapter 11

Numerical and Experimental Study on Contact Force Fluctuation Between Wheel and Rail Considering Rail Flexibility and Track Conditions

Saki Ienaga, Yoshiaki Terumichi, Kazuhiko Nishimura
and Minoru Nishina

Abstract In the present study, we propose an analytical model with a multibody system considering three-dimensional wheel/rail contact geometry and ballasted track characteristics. Suppression of contact force fluctuation between the wheel and rail is desirable from the viewpoint of ensuring running safety, track maintenance, ride comfort, and minimizing the impact of factors such as noise on the surrounding environment. In the present paper, we investigate the effects of the support characteristics of ballasted track on the interaction between vehicles and tracks. Numerical simulations and experiments are carried out for railway vehicle motion under a wide range of ballasted track rigidities. Using the proposed numerical model, we obtain analysis results that are consistent with experimental results under two different track conditions: one simulating ordinary ballasted track characteristics and one that provides sufficient space between sleepers and ballasts. The proposed numerical simulation can accurately analyze vehicle motion running over ballasted track by considering the interaction between the vehicle and the track.

S. Ienaga (✉)

Department of Science and Technology, Graduate School of Sophia University,
7-1 Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
e-mail: saki.ienaga@sophia.ac.jp

Y. Terumichi

Department of Science and Applied Science, Sophia University, 7-1 Kioi-cho,
Chiyoda-ku, Tokyo 102-8554, Japan
e-mail: y-terumi@sophia.ac.jp

K. Nishimura · M. Nishina

Technology Research and Development Department (KOMAKI), Central Japan Railway
Company, 1545-33 Ohyama, Komaki-shi, Aichi 485-0801, Japan
e-mail: kazuhiko.nishimura@jr-central.co.jp

M. Nishina

e-mail: minoru.nishina@jr-central.co.jp

11.1 Introduction

Railways are used worldwide as a transportation system that has minimal impact on the environment. Numerous studies have investigated the running safety and ride comfort of railways. These studies have investigated the coupled vibration of vehicle/track motion based on the contact between wheels and rails [1–6].

In particular, suppressing fluctuations in the contact force between wheels and rails leads to reduced oscillation of vehicles and reduced dynamic loads on the tracks [7, 8]. However, continuous observation of contact force fluctuations generated during the running of railway vehicles and derivation of the vibration characteristics related to the vehicle and track state require a great deal of effort. Interpreting the measurement results for vehicle motion is not easy because there are numerous fluctuation factors involved, vehicle running conditions are not always stable, and limitations exist in performing frequency analyses of the transient response. In contrast, numerical simulations enable us to freely set the conditions for vehicles and tracks and to understand their behavior continuously and simultaneously [9–11]. Therefore, numerical simulations are an effective method of clarifying the mechanism that gives rise to fluctuations in the wheel/rail contact force. Modeling and formulation with high accuracy are important when numerically analyzing the complex motion in the vehicle/track system.

In the present study, we investigate the mechanism of contact force fluctuations through both numerical and experimental approaches. The modeling and formulation of a vehicle and track system are performed considering the three-dimensional wheel/rail contact geometry and the flexibility of the track from the viewpoint of multibody dynamics. Numerical simulations using the proposed model are carried out for vehicle motion under various track-support conditions. Experiments are performed under identical conditions, and the obtained results are used to improve the accuracy of the numerical simulation results.

11.2 Modeling and Formulation

The modeling and formulation for a railway vehicle and track are discussed in this section. In order to construct a multibody system considering three-dimensional motion, the equations of motion are described without simplifications such as a lack of in-plane motion calculations or the linearization of angular displacements. We propose a vehicle/track model that includes the wheel/rail contact model shown in Fig. 11.1.

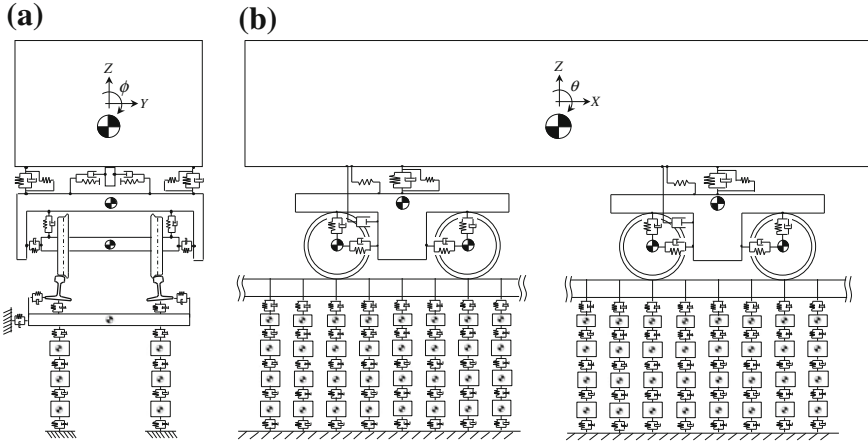


Fig. 11.1 Analytical model. a Front view. b Side view

11.2.1 Vehicle Model

The vehicle model is shown in Fig. 11.2. The vehicle model consists of one car body, two bogie frames, and four wheelsets with two wheels in each. These are regarded as rigid bodies connected by springs and dampers. Thus, the vehicle model is a coupled vibration system of seven rigid bodies. Each body has six degrees of freedom (DOFs): longitudinal displacement X^i , lateral displacement Y^i , vertical displacement Z^i , yaw angle ψ^i , roll angle ϕ^i , and pitch angle θ^i . Angles ϕ^i , θ^i , and ψ^i are rotational angles about the body coordinate axes x^i , y^i , and z^i , which originate at the center of gravity of body i . The superscript indicates the particular body: $i = C, B1, B2, W1, W2, W3, W4$, where C, B, and W refer to the car body, the bogie frame, and the wheelset, respectively. The connecting elements between the bogie frame and the wheelset are eight primary suspensions, each consisting of three springs (k_{1X}, k_{1Y}, k_{1Z}) and three dampers (c_{1X}, c_{1Y}, c_{1Z}). The connecting elements between the car body and the bogie frame are four air suspensions, each consisting of three springs (k_{2X}, k_{2Y}, k_{2Z}) and a damper (c_{2Z}); two lateral stoppers and dampers, each consisting of a spring (k_{3Y}) and a damper (c_{3Y}); four yaw dampers (c_{4X}); and two traction devices, each consisting of two springs (k_{5X}, k_{5Z}). The subscripts X, Y, and Z indicate the longitudinal, lateral, and vertical directions, respectively. The forces of the lateral stopper (k_{3Y}) and the vertical spring of the traction device (k_{5Z}) act as extra forces only if the displacement of the car body relative to the bogie frame exceeds the clearances. To account for the strong nonlinearities of the springs caused by large vehicle motions, modeling of the nonlinearities is introduced as follows:

- (a) With respect to the relative motion of the bogie frame and the wheelset, the longitudinal and lateral stiffness of the primary suspension increases (k_{1X_s}, k_{1Y_s}) if the relative displacement becomes large due to the nonlinearity in the laminated

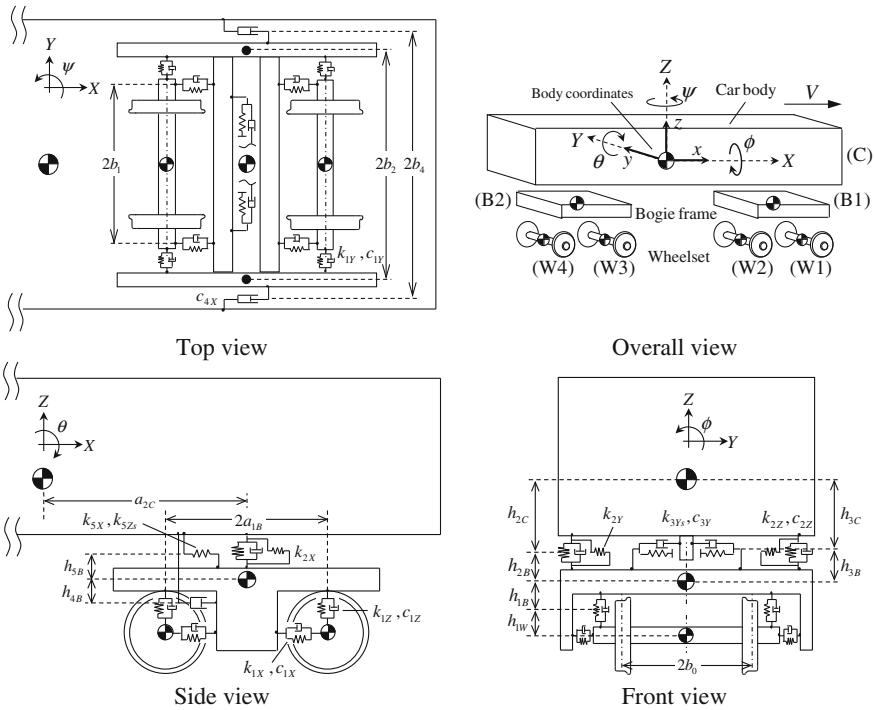


Fig. 11.2 Vehicle model

rubber in the suspension, and the vertical stiffness of the primary suspension increases (k_{1Z}) if the vertical relative displacement is larger than the clearance due to the stopper between the bogie frame and the wheelset;

- (b) With respect to the relative motion of the car body and the bogie frame, the vertical stiffness of the air suspension increases (k_{2Zs}) if the downward relative displacement is larger than the clearance inside the air suspension.

11.2.2 Track Model

The track model is shown in Fig. 11.3. The track model consists of rails, sleepers, and ballasts connected to each other by springs and dampers. In the track model, the rails are treated as flexible beams by applying an absolute nodal coordinate formulation (ANCF) [12–14]. In the ANCF, the global position vector of an arbitrary point on the element can be described using the global shape function and the nodal coordinates as follows:

$$\mathbf{r} = \mathbf{S}(x, y, z)\mathbf{e} \tag{11.1}$$

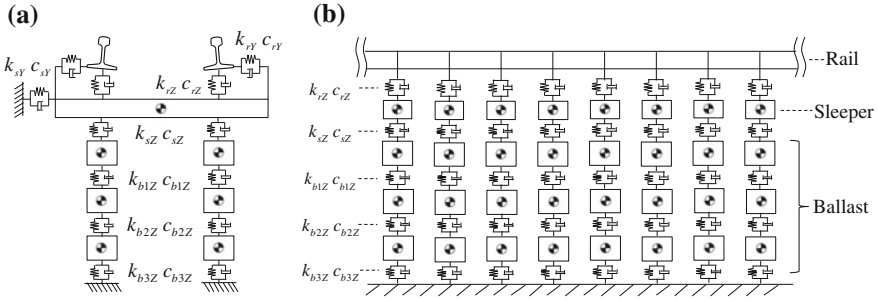


Fig. 11.3 Track model. **a** Front view. **b** Side view

where \mathbf{S} is the global shape function, and \mathbf{e} is the vector of the element nodal coordinates. The global shape function \mathbf{S} is given by

$$\mathbf{S} = [S_1\mathbf{I} \quad S_2\mathbf{I} \quad S_3\mathbf{I} \quad S_4\mathbf{I} \quad S_5\mathbf{I} \quad S_6\mathbf{I} \quad S_7\mathbf{I} \quad S_8\mathbf{I}] \tag{11.2}$$

$$S_1 = 1 - 3\xi^2 + 2\xi^3, \tag{11.3}$$

$$S_2 = l_e (\xi - 2\xi^2 + \xi^3), \tag{11.4}$$

$$S_3 = l_e (\eta - \xi\eta), \tag{11.5}$$

$$S_4 = l_e (\zeta - \xi\zeta), \tag{11.6}$$

$$S_5 = 3\xi^2 - 2\xi^3, \tag{11.7}$$

$$S_6 = l_e (-\xi^2 + \xi^3), \tag{11.8}$$

$$S_7 = l_e \xi \eta, \tag{11.9}$$

$$S_8 = l_e \xi \zeta \tag{11.10}$$

where $\xi = x/l_e$, $\zeta = y/l_e$, and $\eta = z/l_e$ are the displacements of an arbitrary point (x, y, z) from the origin in the undeformed reference configuration, and l_e is the length of the element. The nodal coordinates are defined as nodal displacements and slopes that are determined in the undeformed reference configuration as follows:

$$\mathbf{e} = [(\mathbf{r})^T \quad (\partial\mathbf{r}/\partial x)^T \quad (\partial\mathbf{r}/\partial y)^T \quad (\partial\mathbf{r}/\partial z)^T]^T \tag{11.11}$$

This method yields a constant mass matrix derived from the kinetic energy of the element and a nonlinear stiffness matrix derived from the strain energy.

The rails are supported by concrete sleepers placed at specific intervals. The sleepers are regarded as rigid bodies, each of which has three DOFs: lateral displacement Y^{sl} , vertical displacement Z^{sl} , and roll angle ϕ^{sl} . These bodies are connected to the rails and ballasts elastically. The ballasts are regarded as lumped masses with vertical displacement Z^{ba} and form three layers that are elastically connected to the sleepers and the subgrade.

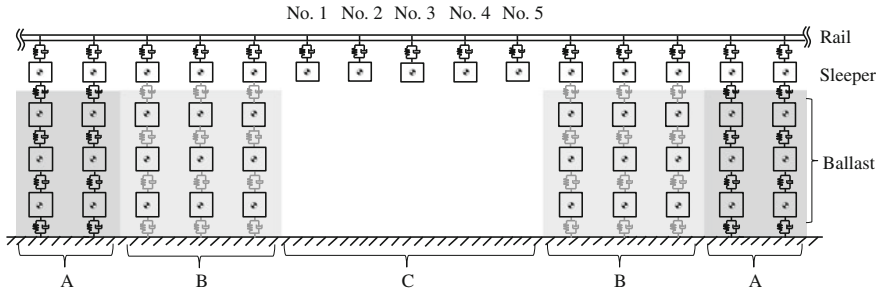


Fig. 11.4 Test track model with unsupported sleepers

11.2.3 Track Model Under a Wide Range of Support Rigidities

In order to investigate the effect of the support rigidity of the ballasted track, a numerical analysis is performed for a vehicle running over track with extreme changes in support stiffness or damping. As shown in Fig. 11.4, we simulate the following three sections of ballasted track support.

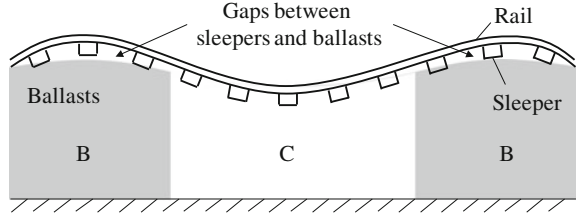
- Section A: The track has regular support rigidity.
- Section B: The boundary area between sections A and C. This section has lower support stiffness and damping than section A.
- Section C: The support stiffness and damping are removed.

In section C, in which the sleepers are completely unsupported by ballasts, track rigidity is reduced. This leads to track irregularity because rail deformation increases due to cyclic loading of the running vehicles. In section B, which is situated before and after section C, rails are deformed in the upper direction, and a gap is generated between sleepers and ballasts, as shown in Fig. 11.5. In this model, the length of section B equivalent to three sleepers, and the support force in section B simulates the gap due to the track irregularity and the reduced track rigidity at the boundary. Moreover, (a) the support force does not act unless the rail displacement reaches the gap size z_d , and (b) the support stiffness and damping coefficients are set to values smaller than those in section A. Therefore, the support force between sleepers and ballasts F_{sZ} and the support force among ballasts F_{bZ} are formalized as follows:

$$F_{sZ} = \begin{cases} 0 & (|z_s - z_{ba}| < z_d) \\ C_B \cdot k_{sZ} \{(z_s - z_{ba}) - z_d\} + C_B \cdot c_{sZ} (\dot{z}_s - \dot{z}_{ba}) & (z_s - z_{ba} \geq z_d) \\ C_B \cdot k_{sZ} \{(z_s - z_{ba}) + z_d\} + C_B \cdot c_{sZ} (\dot{z}_s - \dot{z}_{ba}) & (z_s - z_{ba} \leq -z_d) \end{cases} \quad (11.12)$$

$$F_{bZ} = \begin{cases} 0 & (|z_{ba}| < z_d) \\ C_B \cdot k_{bZ} (z_{ba} - z_d) + C_B \cdot c_{bZ} \dot{z}_{ba} & (z_{ba} \geq z_d) \\ C_B \cdot k_{bZ} (z_{ba} + z_d) + C_B \cdot c_{bZ} \dot{z}_{ba} & (z_{ba} \leq -z_d) \end{cases} \quad (11.13)$$

Fig. 11.5 Track irregularity around section C



where k_{sZ} and c_{sZ} are the stiffness and damping coefficients between the sleeper and the ballast, respectively; k_{bZ} and c_{bZ} are the stiffness and damping coefficients among the ballasts and z_s , respectively; and z_{ba} is the vertical displacement of the sleeper and the ballast. Moreover, C_B is the coefficient of the decrease in the support stiffness and damping and is set to 0.5, and z_d is the length of the gap between sleepers and ballasts in section B and is set to 1.0 mm.

11.2.4 Wheel/Rail Contact Model [15–17]

The normal contact force between the wheel and the rail is defined using Hertz’s contact theory [18] as the elastic contact model. The normal contact force is defined by the amount of elastic deformation of wheel and rail, δ_n . The normal contact force is given as follows:

$$N = -K_c \delta_n^{3/2} - C_c \dot{\delta}_n |\delta_n| \tag{11.14}$$

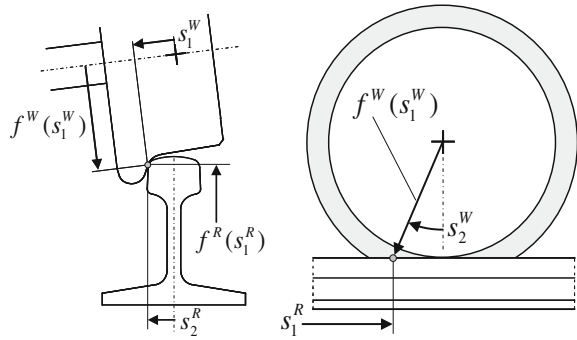
where K_c is the Hertzian contact stiffness, and C_c is the damping coefficient. The damping force is modified by including the factor $|\delta_n|$ so that the contact force is zero when the indentation is zero. The amount of elastic deformation between wheel and rail δ_n is given as

$$\delta_n = \hat{\mathbf{n}}^R \cdot (\mathbf{r}_p^W - \mathbf{r}_p^R) \tag{11.15}$$

where \mathbf{r}_p^W is the global position vector of the contact point on the wheel, \mathbf{r}_p^R is the global position vector of the contact point on the rail, and $\hat{\mathbf{n}}^R$ is the normal unit vector at the contact point on the rail. Tangential forces are defined as creep forces based on a FASTSIM algorithm known as Kalker’s nonlinear theory while the vehicle is in motion [19–21].

In order to precisely calculate the contact force, the three-dimensional contact geometry including the wheel and rail profiles should be considered. The contact geometry is modeled by surface parameters that describe the location of the contact point and define the wheel and the rail profiles. The four parameters for one contact point are as follows: s_1^W is the lateral parameter on the wheel cross section, s_2^W is the circumferential parameter of the wheel, s_1^R is the longitudinal parameter for the rail, and s_2^R is the lateral parameter on the rail cross section. These parameters are

Fig. 11.6 Surface parameters



shown in Fig. 11.6. Using these parameters, the rolling radius (profile of the wheel cross-section) can be written as a function of s_1^W , $f^W(s_1^W)$, and the profile of the rail cross-section can be written as a function of s_2^R , $f^R(s_2^R)$.

$$\mathbf{s} = [s_1^W \quad s_2^W \quad s_2^R \quad s_2^R]^T \tag{11.16}$$

We adopt a contact algorithm that calculates contact points between wheels and rails for every time step in the kinematic analysis. In the contact position analysis, when two rigid bodies come into contact, two types of nonconformal kinematic contact conditions must be satisfied. First, the vector between two contact points on the two surfaces intersects the contact surface at right angles. Second, the two contact surfaces must have the same tangent planes at the contact points. These two conditions define the following four constraint equations, which are required in order to describe the nonconformal contact between the wheel and rail:

$$\mathbf{E}(\mathbf{s}^W, \mathbf{s}^R) = \begin{bmatrix} \mathbf{t}_1^R \cdot (\mathbf{r}_P^W - \mathbf{r}_P^R) \\ \mathbf{t}_2^R \cdot (\mathbf{r}_P^W - \mathbf{r}_P^R) \\ \mathbf{t}_1^W \cdot \mathbf{n}^R \\ \mathbf{t}_2^W \cdot \mathbf{n}^R \end{bmatrix} = \mathbf{0} \tag{11.17}$$

where \mathbf{t}_1^W and \mathbf{t}_2^W are the tangents of the wheel, and \mathbf{n}^R is the normal to the surface of the rail at the contact point. For a given set of wheel/rail generalized coordinates, Eq. (11.17) is solved iteratively by the Newton–Raphson algorithm to determine the surface parameters. This iterative process continues until convergence is achieved.

11.2.5 Formulation of Motion Equations

The equations of motion for the vehicle-track system use the generalized Newton–Euler equations, which are written in matrix form as follows:

$$\mathbf{M}^C \ddot{\mathbf{q}}^C = \mathbf{Q}_f^C + \mathbf{Q}_v^C \quad (11.18)$$

$$\mathbf{M}^{Bi} \ddot{\mathbf{q}}^{Bi} = \mathbf{Q}_f^{Bi} + \mathbf{Q}_v^{Bi} \quad (11.19)$$

$$\mathbf{M}^{Wi} \ddot{\mathbf{q}}^{Wi} = \mathbf{Q}_f^{Wi} + \mathbf{Q}_c^{Wi} + \mathbf{Q}_v^{Wi} \quad (11.20)$$

$$\begin{bmatrix} \mathbf{M}^{Rj} & \mathbf{C}_{e^{Rj}}^{RjT} \\ \mathbf{C}_{e^{Rj}}^{Rj} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{e}^{Rj} \\ \lambda^{Rj} \end{bmatrix} = \begin{bmatrix} -\mathbf{Q}_k^{Rj} - \mathbf{Q}_c^{Rj} - \mathbf{Q}_s^{Rj} + \mathbf{Q}_I^{Rj} \\ \gamma^{Rj} \end{bmatrix} \quad (11.21)$$

$$\mathbf{M}^{Si} \ddot{\mathbf{q}}^{Si} = \mathbf{Q}_s^{Si} + \mathbf{Q}_I^{Si} \quad (11.22)$$

$$\mathbf{M}^{Bai} \ddot{\mathbf{q}}^{Bai} = \mathbf{Q}_s^{Bai} - \mathbf{Q}_I^{Si} \quad (11.23)$$

where superscript i denotes the body number, \mathbf{M}^i is the generalized mass matrix, \mathbf{q}^i is the generalized coordinates, and \mathbf{Q}_v^i is the generalized force vector including inertia forces and gravitational force. The generalized external force vectors are as follows: \mathbf{Q}_f^i is the spring and damper force, \mathbf{Q}_c^i is the wheel/rail contact force, \mathbf{Q}_k^i is the rail elastic force, \mathbf{Q}_s^i is the track support force, and \mathbf{Q}_I^i is the gravitational force. Moreover, $\mathbf{C}_{e^{Rj}}^{Rj}$ is the Jacobian matrix for the constraint of the track support, and λ^{Rj} is the Lagrange multiplier.

11.3 Experiment

In the present study, railway vehicle running experiments are performed under a local change in track rigidity. The wheel/rail contact forces and rail displacements are measured while the vehicle runs over the track.

11.3.1 Experimental Method

Figure 11.7 shows the setup of the experiment. The test line consists of an acceleration section, a test section, and a deceleration section in order to reproduce a real main line. In the test section, sufficient space is provided between the sleepers and ballast-supporting rails and the total length of the space is variable. During the experiment, the test vehicle is pulled from a state of rest by an accelerating truck, and is released by decelerating the truck before the test section.

The test vehicle consists of two real bogies and a car body, on which weights are mounted to adjust the wheel load, and an air tank for operating the air springs in order to reproduce real vehicle motion. However, the distance between the centers of the bogies in the test vehicle is shorter than the distance in an actual vehicle in consideration of the vehicle weight and the truck acceleration performance.

In the experiments, the wheel loads, lateral pressures, and vertical displacements of the rails are measured. The wheel loads and lateral pressures are obtained by converting the strains generated in each wheel. The relation between strain and load

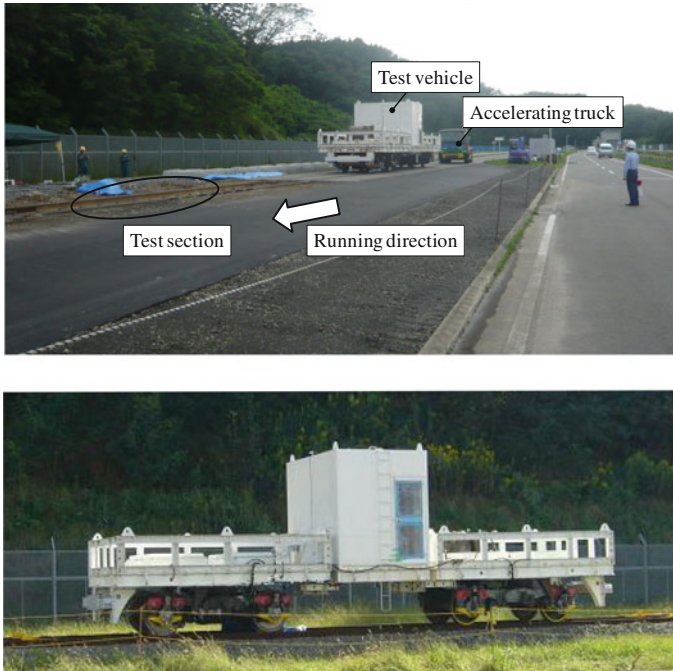


Fig. 11.7 Test track and test vehicle

is known in advance. The vertical displacements of the rails are measured with laser displacement sensors. The sensors are installed in the foundation in such a way as to prevent them from moving with the rails. The sensors measure the amount of displacement by laser irradiation of reflecting plates installed on the rails.

11.3.2 Experimental Results

11.3.2.1 Rail Displacement

Figure 11.8 shows the experimental results for the vertical displacements of the rail under two track support conditions: (a) the vehicle passes over section A, which has regular support rigidity, at 40 km/h, and (b) the vehicle passes over section C, which has five sleepers that are unsupported by ballasts, at 45 km/h.

The time history response waveforms of the rail displacements differ greatly because of the difference in track rigidities. The maximum displacements of the rail at section C are approximately 10 times larger than those at section A.

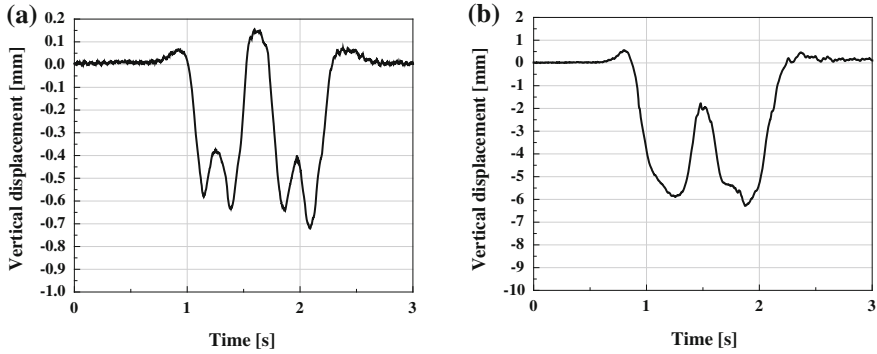


Fig. 11.8 Experimental results for the vertical displacements of the rail. **a** Passing over section A at 40 km/h. **b** Passing over section C at 45 km/h

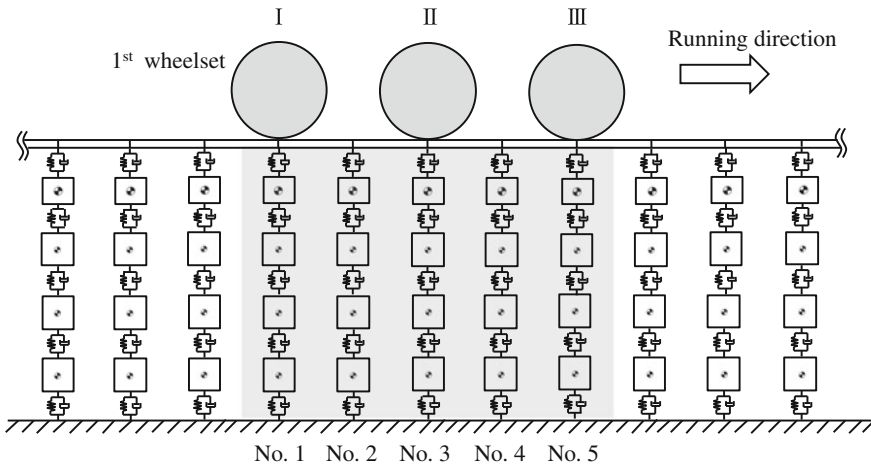


Fig. 11.9 Three positions of the first wheelset on the track

Figure 11.9 shows three positions of the first wheelset on the track. Positions I, II, and III indicate the position at which the first wheelset is located over sleepers Nos. 1, 3, and 5, respectively. Figure 11.10 shows the rail displacement for each sleeper in positions I, II, and III.

The deformation process of the rail as the vehicle travels is shown in Fig. 11.10. For section A, which has regular track rigidity, the shape of the deformation of the rail vary widely with the position of the first wheelset. On the other hand, for section C, which has reduced track rigidity, the deformation shapes of the rail are similar for each position, especially for positions II and III.

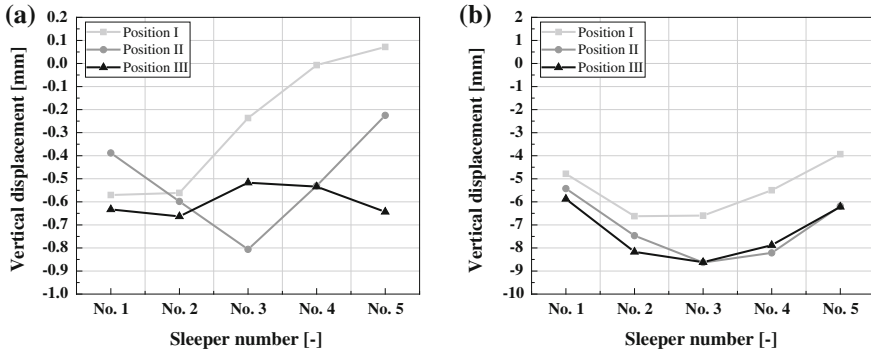


Fig. 11.10 Experimental results for the vertical displacements of rail arrangements I, II, and III. **a** Passing over section A at 40 km/h. **b** Passing over section C at 45 km/h

11.3.2.2 Contact Force

Figure 11.11 shows the experimental results for the wheel load while the vehicle passes over section A, which has a regular support rigidity, and section C, which has three or five sleepers that are unsupported by the ballasts.

Although high frequencies associated with measurement noise and track irregularities are observed in the experimental results, the wheel load fluctuation caused by the change in track rigidity at section C is observed.

In the experimental results shown in Fig. 11.11a, the wheel load caused by the reduced track rigidity of section C changes only slightly. This is because, at low speeds, such as 20 and 25 km/h, the wheel load fluctuates depending primarily on track irregularities.

As shown in Fig. 11.11b, wheel load fluctuations are caused by the reduced track rigidity of section C. The wheel load fluctuates remarkably after section C.

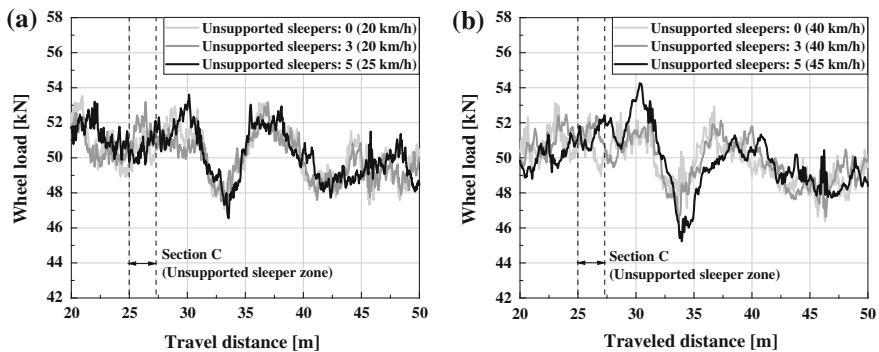


Fig. 11.11 Experimental results for the wheel loads. **a** 20, 25 km/h. **b** 40, 45 km/h

11.4 Numerical Results

In order to investigate the effects of the support characteristics of the ballasted track, numerical simulations are carried out using the proposed analytical model. For the purpose of validation, the numerical results are compared with the results of vehicle running experiments.

11.4.1 Numerical Parameters

Table 11.1 lists the specifications for the numerical simulations. The parameters are based on typical parameters for a Japanese high-speed train. We adopted the arc wheel profile of a Shinkansen and a 60-kg rail profile, both of which are used in the profile functions.

11.4.2 Effect of Track Rigidity on Rail Displacement

The effect of the support characteristics of ballasted track on the rail displacement is discussed in this section.

Figure 11.12 shows the numerical results for the vertical displacements of the rail under two support conditions of the track, in which (a) the vehicle passes over section A at 40 km/h and (b) the vehicle passes over section C, which has five sleepers that are unsupported by ballasts, at 45 km/h. Under both conditions, we obtained numerical results that are consistent with the experimental results shown in Fig. 11.8. Since the track rigidity of section C is lower than that of section A, the following analysis results are obtained. The rail is displaced by approximately 0.7 mm when the wheelsets pass over section A (Fig. 11.12a). On the other hand, the rail is displaced by approximately 6 mm when the front and rear bogies pass over section C (Fig. 11.12b). The numerical results shown in Fig. 11.12 are in good quantitative agreement with the experimental results shown in Fig. 11.8. Furthermore, while the front and rear bogies pass over section A, the rail displacement is restored to the original position at section A (Fig. 11.12a). On the other hand, the rail displacement produced by the front bogie passing over section C is not completely restored (Fig. 11.12b).

Figure 11.13 shows the rail displacement of each sleeper in positions I, II, and III, as shown in Fig. 11.9.

For section C, the rail displacements are approximately 10 times larger than those for section A, which is consistent with the experimental results, as shown in Fig. 11.10. While the first wheelset travels from position II to position III, the rail displacement is not quickly restored because of the reduced track rigidity of section C. This causes the deformation shapes in positions II and III to be similar.

Table 11.1 Numerical parameters

Parameter	Value	Units
Car body mass m_C	31,000	kg
Bogie frame mass m_B	3,200	kg
Wheelset mass m_W	1,800	kg
Longitudinal direction of axle box suspension on bogie frame a_{1B}	1.25	m
Longitudinal direction of air spring on car body a_{2C}	3.75	m
Rail mass per unit length m_R	60.8	kg/m
Sleeper mass m_S	259	kg
Moment of inertia of sleeper I_S	120.1	kg · m ²
Upper ballast m_{Ba1}	59.4	kg
Middle ballast m_{Ba2}	59.4	kg
Lower ballast m_{Ba3}	110.4	kg
Support stiffness of rail (k_{rY}, k_{rZ})	(30, 60)	MN/m
Support damping coefficient of rail (c_{rY}, c_{rZ})	(73, 98)	kN · s/m
Support stiffness of rail k_{sY}	30	MN/m
Support damping coefficient of rail c_{sY}	60	kN · s/m
Support stiffness of rail k_{sZ}	1.19	GN/m
Support damping coefficient of rail c_{sZ}	980	kN · s/m
Support stiffness of rail (k_{b1Z}, k_{b2Z})	(59.3, 54.0)	GN/m
Support damping coefficient of rail (c_{b1Z}, c_{b2Z})	(980, 980)	kN · s/m
Support stiffness of rail k_{b3Z}	49	MN/m
Support damping coefficient of rail c_{b3Z}	980	kN · s/m
Interval of sleeper l_S	0.581	m
Length of track l_T	62.167	m
Number of elements of a rail n_e	108	–
Track gauge $2L_R$	1,435	mm

Although the displacements differ between the numerical and experimental results due to track irregularities, the displacements are in good agreement qualitatively.

11.4.3 Effect of Track Rigidity on Contact Force

The effect of the support characteristics of ballasted track on the vertical contact force, namely wheel load, is discussed in this section.

Figure 11.14 shows the numerical results for the wheel load when the vehicle is passing over section A with regular support rigidity and section C which has three

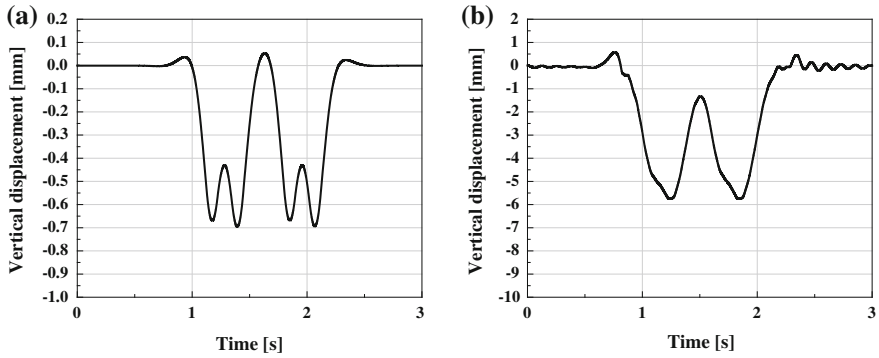


Fig. 11.12 Numerical results for the vertical displacements of the rail. **a** Passing over section A at 40 km/h. **b** Passing over section C at 45 km/h

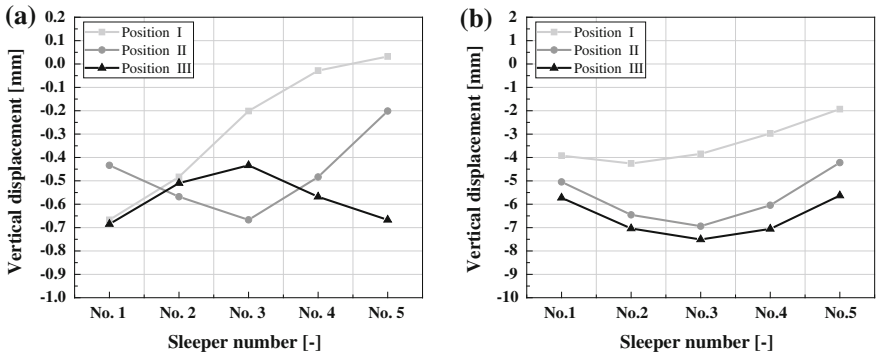


Fig. 11.13 Numerical results for the vertical displacements of rail arrangements I, II, and III. **a** Passing over section A at 40 km/h. **b** Passing over section C at 45 km/h

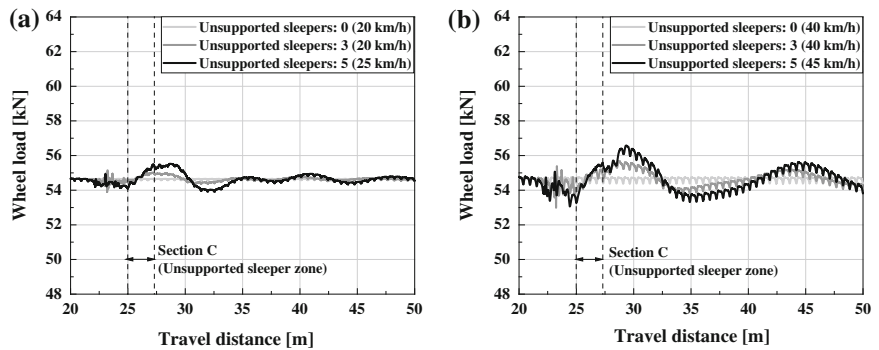


Fig. 11.14 Numerical results for the wheel loads. **a** 20, 25 km/h. **b** 40, 45 km/h

or five sleepers that are unsupported by ballasts. In the numerical simulations, the vehicle speed is set to be the same as in the experiments: 20, 25, 40, and 45 km/h.

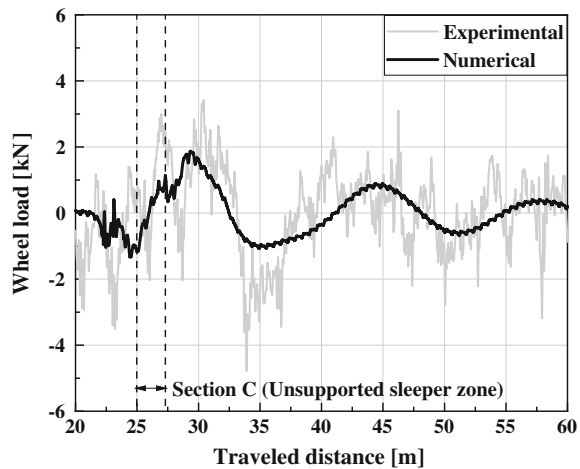
The wheel load fluctuation caused by the change in track rigidity for section C is observed in the numerical results as well as in the experimental results. The details of each analytical condition are described as follows.

The wheel load fluctuates immediately after passing over section C, as shown in Fig. 11.14. And the phase of the wheel load fluctuation is delayed as the number of consecutive unsupported sleepers increases from three to five. The reason for this is that the vehicle performs a pitching motion when it passes section C because of the reduced track rigidity. This result indicates that the pitch angle fluctuation produced by passing section C increases with the increase in the number of unsupported sleepers.

Comparing the numerical results for wheel loads to the experimental results, as shown in Fig. 11.11, the fluctuations in the numerical results are smaller than those in the experimental results because of the track irregularities encountered in the experiments. In order to focus on the effect of the reduced track rigidity in section C, we subtract the results for wheel loads in section A from those in section C, which, in this case, has five unsupported sleepers, as shown in Fig. 11.15.

We can see that, in both the numerical and experimental results at 40 and 45 km/h, the wheel load fluctuates when passing over section C. Although the measurement noises are appeared in the experimental result. However the amounts and the frequencies of the wheel load fluctuations are in good agreement qualitatively.

Fig. 11.15 Experimental and numerical results for the difference between the wheel loads in section C, with 5 unsupported sleepers, and those in section A (45 km/h)



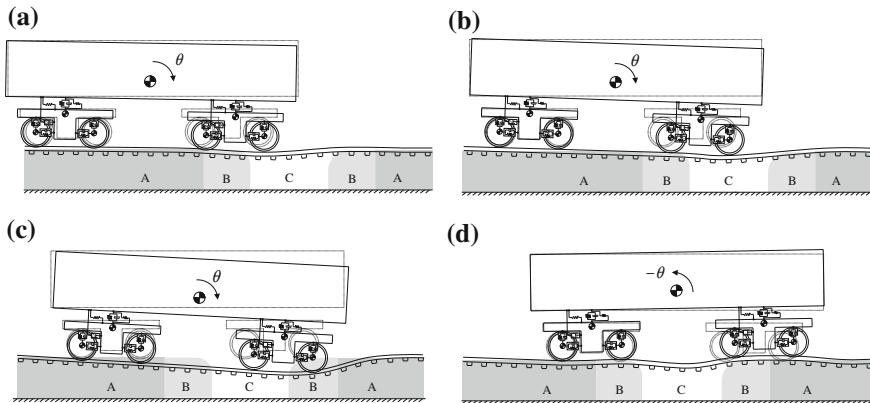


Fig. 11.16 Vehicle motion around section C. **a** The first wheelset has reached section C. **b** The first wheelset is in the middle of section C. **c** The first wheelset has passed section C, but the second wheelset has not. **d** Immediately after the front bogie has passed section C

11.4.4 Effect of Vehicle Motion on Contact Force

In this section, the effect of vehicle motion on the wheel load fluctuation around section C is discussed.

The vehicle performs a pitching motion when it passes section C because of the reduced track rigidity. Figure 11.16a through d show the vehicle motion around section C. When the first wheelset reaches section C, the vehicle starts pitching with a downward motion due to the difference in the track rigidity at the wheel positions. As the pitch angle increases, the wheel load on the first wheelset increases, as shown in Fig. 11.14. After the first wheelset passes section B, following section C, the pitch angle of the vehicle becomes a maximum and the wheel load becomes a maximum (Fig. 11.16c).

11.5 Conclusions

The present study presented a vehicle/track model that takes into account the three-dimensional wheel/rail contact geometry and the ballasted track characteristics. The purpose was to elucidate the effects of the support characteristics of ballasted track on the interaction between the vehicle and the track. Numerical simulations were conducted for a railway vehicle running under the conditions that include a change in track support rigidity, and experiments were conducted in the low-speed range under the same conditions. The following conclusions were obtained for the speed range analyzed in the present study.

- (1) Numerical simulations of the interaction between a passing vehicle and the track can be performed with high accuracy. Good agreement was obtained for the simulations conducted using the proposed model and the experiments under two track conditions: one that simulates regular ballasted track characteristics and one that has sufficient space provided between sleepers and ballasts.
 - (2) An increase in rail displacement occurred when the vehicle passed over the section with reduced track support rigidity. The displacements were not completely restored to their former state between front and rear bogie passes.
 - (3) Although a significant fluctuation in wheel load did not occur when the vehicle passed over the section with the reduced track support rigidity, a rapid wheel load increase was caused by the pitching of the vehicle after passing this section. The amount of wheel load fluctuation increased as the length of the section increased because the pitch angle was larger.
- The vehicle motion when running over ballasted track that has a change in support rigidity was analyzed with high accuracy using the numerical simulation model proposed in the present study.

References

1. Nishimura K, Terumichi Y, Morimura T, Adachi M, Moriahita Y, Miwa M (2015) Using full scale experiments to verify a simulation used to analyze the safety of rail vehicles during large earthquakes. *J Comput Nonlinear Dyn* 10:031013-1
2. Sunami H, Terumichi Y, Morimura T, Adachi M, Moriahita Y (2012) Model for analysis of bogie frame motion under derailment conditions based on full-scale running tests. *Multibody Syst Dyn* 27:321–349
3. Falomi S, Malvezzi M, Meli E (2011) Multibody modeling of railway vehicles: innovative algorithms for the detection of wheel-rail contact points. *Wear* 271:453–461
4. Pombo J, Ambrosio J (2008) Application of a wheel-rail contact model to railway dynamics in small radius curved tracks. *Multibody Syst Dyn* 19:91–114
5. Nielsen JCO (2008) High-frequency vertical wheel-rail contact forces—validation of a prediction model by field testing. *Wear* 265:1465–1471
6. Kalker JJ (1979) Survey of wheel-rail rolling contact theory. *Veh Syst Dyn* 8:317–358
7. Adachi M, Matsumoto A (2011) Improvement of curving performance by expansion of gauge widening and additional measures. In: *Proceedings of the Institution of mechanical engineers, part F: Journal of Rail and Rapid Transit* 0954409711408974
8. Gullers P, Andersson L, Lunden R (2008) High-frequency vertical wheel-rail contact forces—field measurements and influence of track irregularities. *Wear* 265:1472–1478
9. Berggren EG, Li MXD, Spannar J (2006) A new approach to the analysis and presentation of vertical track geometry quality and rail roughness with focus on train-track interaction and wavelength contact. In: *Proceedings of the 7th international conference on contact mechanics and wear of rail/wheel systems*, pp 24–27, Brisbane
10. Jin X, Xiao X, Wen Z, Zhou Z (2006) Effect of sleeper pitch on rail corrugation at a tangent track in vehicle hunting. In: *Proceedings of the 7th international conference on contact mechanics and wear of rail/wheel systems*, pp 179–188, Brisbane
11. Magheri S, Malvezzi M, Meli E, Rindi A (2011) An innovative wheel-rail contact model for multibody applications. *Wear* 271:462–471
12. Shabana AA et al (1998) Application of absolute nodal coordinate formulation to larger rotation and large deformation problem. *ASME J Mech Des* 120:188–195

13. Takahashi Y, Shimizu N (1999) Study on elastic forces of the absolute nodal coordinate formulation for deformable beams. In: ASME proceedings of design engineering technical conference, pp 33–40
14. Sopanen JT, Mikkola AM (2003) Description of elastic forces in absolute nodal coordinate formulation. *Nonlinear Dyn* 34:53–74
15. Shabana AA, Zaazaa KE, Escalona JL, Sany JR (2004) Development of elastic force model for wheel/rail contact problem. *J Sound Vib* 269:295–325
16. Sugiyama H, Araki K, Suda Y (2009) Development of wheel/rail contact procedure in multi-body railroad vehicle dynamics. *Trans Jpn Soc Mech Eng Ser C* 75–751:95–103
17. Shabana AA, Sany JR (2001) An augmented formulation for mechanical systems with non-generalized coordinates: application to rigid body contact problems. *Nonlinear Dyn* 24:183–204
18. Hertz H (1882) Ueber die Verdunstung der Flüssigkeiten, insbesondere des Quecksilbers, im luftleeren Raume. *Ann Phys* 253:177–193
19. Kalker JJ (1982) A fast algorithm for the simplified theory of rolling contact. *Veh Syst Dyn* 11:1–13
20. Kalker JJ (1991) Wheel-rail rolling contact theory. *Wear* 144:243–261
21. Kalker JJ (1973) Simplified theory of rolling contact. *Delft Prog Rep* 1:1–10

Chapter 12

Use of Flexible Models in Extended Kalman Filtering Applied to Vehicle Body Force Estimation

Sebastiaan van Aalst, Frank Naets, Johan Theunissen
and Wim Desmet

Abstract Accurate knowledge of wheel loads is of great value in vehicle design and control. However, a direct measurement of these forces is generally not feasible. This motivates the use of model-based estimation techniques, such as the Kalman filter to obtain operational wheel forces. The general approach in literature is to use simple ad-hoc models (like the bicycle model) in the Kalman filter. In many applications however, including vehicle dynamics, this results in a system that is not observable for all the variables of interest, e.g. the individual tyre forces. In this light, this work proposes the use of general flexible multibody models for Kalman filtering. The introduction of flexible deformations in the model enables the observation of variables which cannot be obtained from a rigid model. This allows the filter to differentiate between the contributions of different input forces. This approach is demonstrated by employing an augmented extended Kalman filter to perform a combined estimation of the current vehicle state and wheel forces of a 2D vehicle model. The system is modeled in a floating-frame-of-reference (FFR) approach and the vehicle body is described by a reduced order finite element model. An observability analysis is performed and the observability conditions for the unknown input forces are derived. The proposed approach is validated numerically and compared to an estimator with a rigid assumption.

S. van Aalst (✉) · F. Naets · W. Desmet
Department of Mechanical Engineering, KU Leuven - University of Leuven,
Celestijnenlaan 300, 3000 Leuven, Belgium
e-mail: sebastiaan.vanaalst@kuleuven.be

F. Naets
e-mail: frank.naets@kuleuven.be

W. Desmet
e-mail: wim.desmet@kuleuven.be

S. van Aalst · J. Theunissen
Flanders Make, Oude Diestersebaan 133, 3920 Lommel, Belgium
e-mail: jtheunissen@tenneco.com

12.1 Introduction

The forces acting on a vehicle are an important area in vehicle dynamics research. Together with the vehicle state, they are essential for determining the dynamic behavior of the vehicle. Accurate knowledge of the external forces, mostly tyre forces, is of specific interest during the vehicle design phase, and is of great value for vehicle control systems that attempt to improve vehicle handling and safety. A complete measurement of the forces and states of interest, however, requires an expensive and complicated sensor installation which is not feasible for mass-produced vehicles. This motivates the use of model-based estimation techniques such as the Kalman filter [1] and its many variations [2] to leverage a limited set of measurements for more information.

The standard Kalman filter assumes that the inputs to the system are known, but in many application this is not the case. For example, in vehicle dynamics applications the wheel loads are typically unknown. In order to perform state estimation in the presence of unknown inputs so-called unbiased Kalman filters have been proposed [3, 4]. However, often there is a special interest to obtain an estimate of the unknown inputs to the system as well. Several variations of the standard Kalman filter exist that can perform a combined estimation of the states and unknown inputs. In this work the augmented (joint) Kalman filter is employed [5]. This approach adds the unknown inputs to the state vector and treats them as additional states to be estimated. A model for the unknown inputs has to be provided.

The estimation accuracy of derived variables, like the wheel loads, is directly related to the accuracy of the model. Currently, the general approach is still to use simple ad-hoc models for the Kalman filter. This limits the possible estimation accuracy, and in many applications, including vehicle dynamics, this results in an unobservable system, especially in the case of multiple unknown inputs. For example, consider the rigid four-wheel vehicle model. A seven degree-of-freedom model can be used with an augmented Kalman filter for state/input estimation [6–8]. With this model, there are eight unknown input forces: a longitudinal and a lateral tyre force at each wheel. Taking into account the measurements that are typically available in a standard car (six degree of freedom inertial measurement unit, wheel speed sensors, steering angle sensor and GPS), an observability analysis will show that the individual tyre forces are not observable. This means that we are not able to obtain valid estimates for these forces. In literature this issue is typically circumvented by making assumptions about the tyre forces such as a known tyre load distribution or by assuming a tyre force model with known parameters. This reduces the number of independent variables that needs to be estimated as such that an observable system is obtained. These assumptions reduce the robustness and the accuracy of the estimation because this information is generally not available and will thus introduce a bias. This issue does not only appear in vehicle dynamics but also arises in many other domains of mechanical input estimation.

Multibody simulation provides a general framework to develop high fidelity models for mechanical systems [9]. Rigid multibody models have already been used for state estimation [10, 11] and for state/input estimation [12]. However, even an extensive rigid multibody model does not necessarily prevent the observability issues which arise when multiple unknown input forces are present. This work demonstrates that introducing flexibility in the bodies for the model is key for obtaining an observable system. Flexibility adds a sufficient number of degrees-of-freedom to differentiate between the contributions of different input forces. The required information can be extracted from additional measurements, such as strain measurements. As a result an observable system can be obtained. A similar approach is followed in structural dynamics applications to estimate external forces applied to the structure [13–15]. The difference being that in a multibody system the flexible body can also undergo large translational and rotational motion, resulting in a nonlinear model. Recently nonlinear model reduction techniques for flexible multibody systems have been developed [16]. These techniques can allow to run these models in real-time together with a Kalman filter.

This paper demonstrates the use of flexible multibody models for estimation by employing the augmented Kalman filter to perform a combined estimation of the current vehicle state and tyre forces for 2D vehicle models (which can be used for estimation longitudinal-vertical, longitudinal-lateral or lateral-vertical models). However, the presented approach can be readily extended to three-dimensional dynamics. The remainder of this paper is divided into three sections: Sect. 12.2 discusses the derivation of the estimation model. This section also discusses the use of a reduced order finite element model to describe the flexible behavior of the vehicle body. Section 12.3 introduces the discrete-time augmented extended Kalman filter and discusses the required derivatives of the model. Here, also an observability analysis is performed and the results are discussed. Finally in Sect. 12.4 the proposed approach is numerically validated and compared to the current general approach for a vehicle model consisting of a flexible vehicle body suspended on a spring-damper suspension system.

12.2 Reduced Order Model for State/Input Estimation

The equations of motion for a deformable body undergoing large translational and rotational motion in a floating-frame-of-reference (FFR) formulation can be summarized as [17]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{g}_{gyr}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}_{ext}(\mathbf{q}, t) \quad (12.1)$$

In this equation \mathbf{q} is the vector of time-dependent generalized coordinates of the system, and \mathbf{M} , \mathbf{C} and \mathbf{K} are respectively the mass-, damping- and stiffness matrix. \mathbf{g}_{gyr} is a quadratic velocity vector that contains the centrifugal and Coriolis force components and \mathbf{g}_{ext} is the vector of generalized external forces. In the FFR formulation the

motion of the body is separated into a large motion of the body reference frame and a linear elastic deformation with respect to the reference frame. The configuration of the deformable body is then defined by a coupled set of generalized reference and elastic coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_f \end{bmatrix} \quad (12.2)$$

In which \mathbf{q}_r and \mathbf{q}_f are the generalized reference coordinates and \mathbf{q}_f is the vector of generalized elastic coordinates. \mathbf{q}_r is the global position of the origin of the body reference frame and \mathbf{q}_f is a set of n_r rotational parameters that describes the orientation of the body reference frame with respect to the global frame.

For the sake of simplicity, this paper only treats the planar case but the presented results can easily be extended to the spatial case. Henceforth \mathbf{q}_r is written as θ .

In what follows, the derivation of \mathbf{g}_{ext} is provided. For the derivation of the full model (12.1) the reader is referred to [17].

Assume that the external force \mathbf{f}_{ext} is applied at an arbitrary point P of the deformable body. The global position of point P can be expressed in function of the generalized coordinates as:

$$\mathbf{r}_p = \mathbf{q}_r + \mathbf{A}(\theta) (\bar{\mathbf{u}}_{0_p} + \mathbf{N}_p \mathbf{q}_f) \quad (12.3)$$

where $\bar{\mathbf{u}}_{0_p}$ is the local position vector of point P in the undeformed state and the term $\mathbf{N}_p \mathbf{q}_f$ describes the local linear elastic deformation. \mathbf{N}_p is a partition of a shape matrix that can be obtained by using e.g. the finite element method. \mathbf{A} is a transformation matrix that transforms the local vector coordinates to the global frame. The generalized external force vector \mathbf{g}_{ext} is then obtained from the virtual work performed by an infinitesimal displacement of the application point \mathbf{r}_p with respect to the generalized coordinates:

$$\mathbf{g}_{ext}(\mathbf{q}, t) = \mathbf{L}^T(\mathbf{q}) \mathbf{f}_{ext}(t) = \left[\mathbf{I} \frac{\partial \mathbf{A}(\theta)}{\partial \theta} (\bar{\mathbf{u}}_{0_p} + \mathbf{N}_p \mathbf{q}_f) \quad \mathbf{A}(\theta) \mathbf{N}_p \right]^T \mathbf{f}_{ext}(t) \quad (12.4)$$

In which the matrix \mathbf{L} describes how the external force loads the generalized coordinates. For estimation purposes in automotive applications it often makes more sense to express the external forces with respect to the body reference frame. This is achieved by substituting in (12.4) the external forces by the product of the transformation matrix \mathbf{A} and the external forces expressed with respect to the body reference frame:

$$\mathbf{g}_{ext}(\mathbf{q}, t) = \bar{\mathbf{L}}^T(\mathbf{q}) \bar{\mathbf{f}}_{ext}(t) = \left[\mathbf{A}^T(\theta) \quad \widetilde{(\bar{\mathbf{u}}_{0_p} + \mathbf{N}_p \mathbf{q}_f)} \quad \mathbf{N}_p \right]^T \bar{\mathbf{f}}_{ext}(t) \quad (12.5)$$

where the tilde denotes the skew symmetric transformation.

The remainder of this section consists of three subsections. The first subsection discusses the use of a reduced order finite element model to describe the behavior of the deformable body. The second subsection discusses a technique called state augmentation, which allows to obtain estimates of the unknown external forces in a Kalman filtering approach. The third subsection discusses the time-discretization of the equations of motion, which is required to employ the discrete-time extended Kalman filter.

12.2.1 *Model Order Reduction by Modal Truncation Augmentation*

The flexible behavior of a body can be modeled using the finite element method [18]. However, if the goal is to perform the model-based estimation in an online application, it is typically not feasible to use a full finite element model. This is due to the large amount of degrees-of-freedom (DOFs) and the high-frequency content of these models, which leads to large computational loads. Therefore, a reduced order model should be employed. Over the past decades, several model order reduction techniques have been proposed [19] for linear mechanical systems, which can be employed on a component-level in a FFR model. Historically, the most widely used reduction technique for structural systems is modal reduction, which is based on a truncated modal expansion of the system. However, this technique exhibits poor convergence because the modes retained are usually not sufficient to represent the spatial distribution of the applied load. This shortcoming can be overcome by adding to the set of retained modes additional component modes that take the specific loading situation into account. This technique is applied in this work to reduce a finite element model of the body. A statically complete dynamic mode superset is employed:

$$\Psi = [\Psi_{rb} \ \Psi_{ff} \ \Psi_{rira}] \quad (12.6)$$

where Ψ_{rb} are the rigid-body modes, Ψ_{ff} is the set of retained free-free normal modes, and Ψ_{rira} are the so-called residual inertia-relief attachment modes [20]. An attachment mode is defined as the static deformation of the system for a unit input force at a specific DOF. Since in this work, the body has rigid-body freedom, the attachment modes are made inertia-relief. This comes down to equilibrating the applied unit force by the d'Alembert forces associated with the rigid-body acceleration. Furthermore, the attachment modes are residualized such that they are guaranteed to be linearly independent of the set of retained free-free normal modes.

The reduction of the finite element model is performed by representing the component's physical displacement DOFs \mathbf{d} , in terms of the generalized coordinates \mathbf{p} :

$$\mathbf{d} = \Psi \mathbf{p} \quad (12.7)$$

In which the number of modes in Ψ is typically much less than the number of DOFs of the original system. The resulting reduced finite element model is then used to construct the model in (12.1) according to the FFR formulation.

12.2.2 State Augmentation for Input Estimation

In order to use model (12.1) in a Kalman filtering approach, it has to be transformed into first-order form:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \bar{\mathbf{f}}_{ext}) \quad (12.8)$$

The state-space coordinates vector \mathbf{x} for this system is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (12.9)$$

and the corresponding continuous-time system equations are:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\bar{\mathbf{L}}^T \end{bmatrix} \bar{\mathbf{f}}_{ext} - \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{g}_{gyr} \end{bmatrix} \quad (12.10)$$

where for ease of notation, the time-dependency of $\bar{\mathbf{f}}_{ext}$, and the dependency of \mathbf{M} , $\bar{\mathbf{L}}$ and \mathbf{g}_{gyr} on \mathbf{q} and $\dot{\mathbf{q}}$ are omitted from the notation. These equations can be summarized as:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\bar{\mathbf{f}}_{ext} - \mathbf{c}(\mathbf{x}) \quad (12.11)$$

For estimating the input simultaneously with the system state in a Kalman filtering approach, a technique called state augmentation is employed [5]. In this approach the unknown input forces are added to the state vector and are treated as additional states to be estimated. In this work it is assumed that all the input forces are unknown. This leads to the augmented state vector \mathbf{x}^* :

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x} \\ \bar{\mathbf{f}}_{ext} \end{bmatrix} \quad (12.12)$$

A model for the unknown forces has to be provided in this case as well. Here, a zeroth-order random walk model is employed for each unknown force. This model assumes the force to be constant except for an additive uncertainty:

$$\dot{\bar{\mathbf{f}}}_{ext} = \mathbf{0} + \mathbf{r}_f \quad (12.13)$$

where \mathbf{r}_f is continuous-time noise, which indicates that the rate of change of the external force is expected to be a random process. This approach has been applied

with success by Naets [13, 14, 21] and Lourens [15] and higher order versions of this approach have been used by Ray [6, 7]. In general, \mathbf{r}_f is not exactly known and serves as a tuning parameter for the Kalman filter. Since the external forces can show strong variations, \mathbf{r}_f is typically quite high. The augmented continuous-time system is:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\bar{\mathbf{f}}}_{ext} \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{x}) & \mathbf{B}(\mathbf{x}) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \bar{\mathbf{f}}_{ext} \end{bmatrix} - \begin{bmatrix} \mathbf{c}(\mathbf{x}) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_f \end{bmatrix} \quad (12.14)$$

which can be summarized as:

$$\dot{\mathbf{x}}^* = \mathbf{A}^*(\mathbf{x}) \mathbf{x}^* - \mathbf{c}^*(\mathbf{x}) + \mathbf{r}_{x^*} \quad (12.15)$$

$$= \mathbf{g}^*(\mathbf{x}^*) + \mathbf{r}_{x^*} \quad (12.16)$$

where \mathbf{r}_{x^*} is the continuous-time model noise vector with associated covariance matrix \mathbf{Q}^* . Notice that the noise on the continuous-time multibody model is assumed to be zero and all noise is assumed on the unknown forces. This is a valid approximation as unknown forces generally dominate the uncertainty in the model. It is important to mention that this system is nonlinear. Both the matrix \mathbf{A}^* and the vector \mathbf{c}^* depend on the system state \mathbf{x} . A nonlinear Kalman filter, like the extended Kalman filter, is required to handle this kind of system. This approach is chosen in this work due to the low computational load, but other schemes like unscented Kalman filters can be used as well.

12.2.3 Discretization of Equations of Motion

In this work the discrete-time extended Kalman filter is employed because it is particularly suitable for efficient computer implementation. In order to apply this filter to system (12.16), the equations of motion need to be time-discretized. Several integration schemes exist to discretize these equations. In this work an exponential Euler integration scheme is chosen with zero-order hold for the inputs [22]. This approach allows for larger time steps than typical explicit integrators. The following discrete-time equations of motion are obtained:

$$\begin{bmatrix} \mathbf{x} \\ \bar{\mathbf{f}}_{ext} \end{bmatrix}_k = \begin{bmatrix} e^{\mathbf{A}\Delta t} & \mathbf{A}^{-1} (e^{\mathbf{A}\Delta t} - \mathbf{I}) \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}_{k-1} \begin{bmatrix} \mathbf{x} \\ \bar{\mathbf{f}}_{ext} \end{bmatrix}_{k-1} - \begin{bmatrix} \mathbf{A}^{-1} (e^{\mathbf{A}\Delta t} - \mathbf{I}) \mathbf{c} \\ \mathbf{0} \end{bmatrix}_{k-1} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_f \end{bmatrix}_{k-1} \quad (12.17)$$

where Δt is the time step size and subscript k refers to the time sample k . For ease of notation, the dependency of \mathbf{A} , \mathbf{B} and \mathbf{c} on \mathbf{x} is omitted from the notation. These

equations can be summarized as:

$$\mathbf{x}_k^* = \mathbf{A}_d(\mathbf{x}_{k-1}) \mathbf{x}_{k-1}^* - \mathbf{c}_d(\mathbf{x}_{k-1}) + \mathbf{r}_{x_{k-1}^*} \quad (12.18)$$

$$= \mathbf{g}_d(\mathbf{x}_{k-1}^*) + \mathbf{r}_{x_{k-1}^*} \quad (12.19)$$

where $\mathbf{r}_{x_k^*}$ is the discrete-time process noise vector with associated covariance matrix \mathbf{Q}_k^* :

$$\mathbf{Q}_{k-1}^* = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}^*(\mathbf{x})(t_k-\tau)} \mathbf{Q}^* e^{\mathbf{A}^{*T}(\mathbf{x})(t_k-\tau)} d\tau \quad (12.20)$$

Solving this integral is not straightforward, however, \mathbf{Q}_k^* can also be calculated by first constructing a matrix, and then computing the matrix exponential of it. More details about the computation of \mathbf{Q}_k^* can be found in [23].

The discrete-time equations of motion (12.19) and the covariance matrix \mathbf{Q}_k^* can be inserted in the discrete-time augmented extended Kalman filter to perform the coupled state/input estimation.

12.3 The Augmented Extended Kalman Filter for Coupled State/Input Estimation

As mentioned before, the system considered is nonlinear. In this work, the extended Kalman filter is selected to perform the estimation. This filter is a straightforward extension of the original linear Kalman filter to nonlinear systems [2]. The system equations of motion (12.19) are complemented by the (nonlinear) measurement equations:

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k^*) + \mathbf{r}_y \quad (12.21)$$

in which \mathbf{y}_k contains the sensor measurements as obtained by the (nonlinear) measurement equations \mathbf{h} and the measurement noise \mathbf{r}_y . The noise is assumed to be zero-mean uncorrelated Gaussian noise with covariance matrix \mathbf{R} . Just like the original linear Kalman filter, the extended Kalman filter algorithm is recursive and operates in two steps: a time update step and a measurement update step. The time update step predicts the current system state $\hat{\mathbf{x}}_k^{*-}$ and propagates the state error covariance matrix \mathbf{P}_k^- :

$$\hat{\mathbf{x}}_k^{*-} = \mathbf{g}_d(\hat{\mathbf{x}}_{k-1}^{*+}) \quad (12.22)$$

$$\mathbf{P}_k^- = \mathbf{G}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{G}_{k-1}^T + \mathbf{Q}_{k-1}^* \quad (12.23)$$

Once the measurement \mathbf{y}_k is available, the Kalman gain \mathbf{K}_k is calculated which is then used to update the system state estimate $\hat{\mathbf{x}}_k^{*+}$ by incorporating the information available in the measurement. The updated state error covariance matrix \mathbf{P}_k^+ is also computed:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (12.24)$$

$$\hat{\mathbf{x}}_k^{*+} = \hat{\mathbf{x}}_k^{*-} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^{*-})) \quad (12.25)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (12.26)$$

In both the time- and measurement update step, the propagation of the state error covariance is performed by linearizing, respectively, the system- and measurement equations around the current configuration:

$$\mathbf{G}_{k-1} = \frac{\partial \mathbf{g}_d}{\partial \mathbf{x}^*} (\mathbf{x}_{k-1}^*) \quad (12.27)$$

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}^*} (\mathbf{x}_k^*) \quad (12.28)$$

Calculating \mathbf{H} is usually rather straightforward. Calculating \mathbf{G} , however, the derivative of the equations of motion, is more involved. This derivative is elaborated in the next subsection.

12.3.1 Required Derivatives

In order to employ the extended Kalman filter, the derivative of the discrete-time equations of motion with respect to the augmented state vector is required. This derivative is a square matrix in which each column j is the derivative of the equations of motion with respect to augmented state x^{*j} :

$$\mathbf{G}_{k-1}^j = \frac{\partial \mathbf{g}_d}{\partial x^{*j}} (\mathbf{x}_{k-1}^*) = \mathbf{A}_d^j (\mathbf{x}_{k-1}) + \frac{\partial \mathbf{A}_d}{\partial x^{*j}} (\mathbf{x}_{k-1}) \mathbf{x}_{k-1}^* - \frac{\partial \mathbf{c}_d}{\partial x^{*j}} (\mathbf{x}_{k-1}) \quad (12.29)$$

for $j = 1 \dots n^*$, where n^* is the number of augmented states. In order to evaluate (12.29) the following expressions must be used:

$$\frac{\partial \mathbf{A}}{\partial x^{*j}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial x^{*j}} & \mathbf{M}^{-1} \mathbf{K} & \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial x^{*j}} & \mathbf{M}^{-1} \mathbf{C} \end{bmatrix} \quad (12.30)$$

$$\frac{\partial \mathbf{B}}{\partial x^{*j}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \frac{\partial \bar{\mathbf{L}}^T}{\partial x^{*j}} - \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial x^{*j}} & \mathbf{M}^{-1} \bar{\mathbf{L}}^T \end{bmatrix} \quad (12.31)$$

$$\frac{\partial \mathbf{c}}{\partial x^{*j}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \frac{\partial \mathbf{g}_{gyr}}{\partial x^{*j}} - \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial x^{*j}} & \mathbf{M}^{-1} \mathbf{g}_{gyr} \end{bmatrix} \quad (12.32)$$

For the derivative of the mass matrix \mathbf{M} and of the quadratic velocity vector \mathbf{g}_{gyr} the reader is referred to [17]. The derivative of matrix $\bar{\mathbf{L}}$ can be obtained rather straightforward from (12.5).

12.3.2 Observability

The observability for a given set of measurements is a very important system property. Therefore, before moving on to the numerical validation, an observability analysis is performed. A system is said to be observable if its state vector at a certain time instant can be uniquely determined given a finite sequence of its outputs. Less formally, observability means that the measurements can provide sufficient information for estimating the state vector. Observability is thus a property of a certain system-measurement combination. In this work four different kinds of measurements are considered:

- **Position measurement:** For a position tracking of point P , e.g., using an optical tracking system, the measurement equation is:

$$\mathbf{r}_p = \mathbf{q}_t + \mathbf{A}(\theta) \bar{\mathbf{u}}_p, \quad \bar{\mathbf{u}}_p = \bar{\mathbf{u}}_{0_p} + \Psi_{p_{trans}} \mathbf{q}_f \quad (12.33)$$

where $\bar{\mathbf{u}}_p$ is the local position vector of point P , and $\Psi_{p_{trans}}$ is the partition of the modeset (12.6) associated to the translational DOFs of point P .

- **Acceleration measurement:** For an accelerometer at point P , the measurement equation is:

$$\bar{\mathbf{a}}_p = \mathbf{A}^T(\theta) \ddot{\mathbf{q}}_t + \bar{\boldsymbol{\omega}} \times (\bar{\boldsymbol{\omega}} \times \bar{\mathbf{u}}_p) + \bar{\boldsymbol{\alpha}} \times \bar{\mathbf{u}}_p + 2\bar{\boldsymbol{\omega}} \times \dot{\bar{\mathbf{u}}}_p + \ddot{\bar{\mathbf{u}}}_p \quad (12.34)$$

where $\bar{\boldsymbol{\omega}}$ and $\bar{\boldsymbol{\alpha}}$ are, respectively, the angular velocity vector and angular acceleration vector defined in the body reference. The expression for $\bar{\mathbf{a}}_p$ can be rewritten such that all variables are expressed as function of the system state.

- **Angular velocity measurement:** For a gyroscope at point P , the measurement equation is:

$$\bar{\boldsymbol{\omega}}_p = \bar{\boldsymbol{\omega}} + \Psi_{p_{rot}} \dot{\mathbf{q}}_f \quad (12.35)$$

where $\Psi_{p_{rot}}$ is the partition of the modeset (12.6) associated to the rotational DOFs of point P .

- **Strain evaluation:** For a strain gauge at point P , the measurement equation is:

$$\varepsilon = \mathbf{BL}\Psi_p \mathbf{q}_f \quad (12.36)$$

where \mathbf{B} is a boolean matrix that selects the measured strain component, and \mathbf{L} is a matrix that describes the strain-displacement relationship using the finite element shape functions.

As mentioned before, the model and measurements considered are nonlinear. Determining the observability for a nonlinear system is substantially more difficult than for a linear system. An approximate method of checking the observability of nonlinear systems is to perform a linearized observability analysis. This can already provide some general conditions for observability. In this work the Popov-Belevitch-Hautus (PBH) test of observability of a linear system is applied on the linearized continuous-time system [24]. This test states that the linearized system is observable if and only if:

$$\text{rank} \begin{bmatrix} \mathbf{G}_c - s\mathbf{I} \\ \mathbf{H} \end{bmatrix} = n,^* \text{ for all } s \in \mathbb{C} \quad (12.37)$$

where \mathbf{G}_c is the derivative of the continuous-time system and \mathbf{H} is the derivative of the measurement equations. It is sufficient to check the rank (12.37) for values of s equal to the eigenvalues of the matrix \mathbf{G}_c since for all other values of s the matrix $\mathbf{G}_c - s\mathbf{I}$ is of full rank by definition of the eigenvalue-problem. Here, only $s = 0$ is investigated because this is where the problems typically occur for force estimation [13]. In this case, the following considerations can be made:

- the number of DOFs of the reduced finite element model should be greater than or equal to the number of forces to be estimated. This condition indicates that not only accuracy requirements can dictate the size of the reduced order model, but also observability must be taken into account.
- the number of strain measurements should be greater than or equal to the number of forces to be estimated. If this is not satisfied the filter cannot differentiate between the contributions of different input forces.
- at least one global x - and y -position measurement should be performed. This measurement is required to observe the states associated to the reference coordinates.

These considerations will be taken into account for the numerical validation in Sect. 12.4. Notice that it is not required to perform acceleration or angular velocity measurements. However, these measurements do provide valuable information that can lead to a better estimation accuracy. A further analysis of (12.37) and of the model and measurements considered, reveals that it is possible to observe the unknown forces by performing the strain measurements only. In this case, the states associated to the reference coordinates are not observable but in this work the main interest is in the unknown input forces such that these variables are not of specific interest. This is a very good feature since global position measurements are difficult to obtain in practice.

It is interesting to perform a similar observability analysis in case of a rigid model. This corresponds to the current general approach. In this case, the information available in strain measurements cannot be extracted because the model does not contain this effect. Making use of criterion (12.37) it can be shown that a maximum

of three unknown external forces can be estimated, requiring at least one angular velocity measurement and a longitudinal and lateral acceleration measurement. This demonstrates the benefit of introducing flexible deformations in the model. It enables the observation of variables that cannot be obtained from a rigid model. This allows the filter to differentiate between the contributions of different input forces.

12.4 Numerical Validation

In order to validate the proposed approach, a numerical experiment is performed in MATLAB. The proposed approach is compared to the current general approach where the flexibility of the body is not taken into account in the model for the Kalman filter. The first subsection introduces the system and describes the models used. The second subsection shows and discusses the simulation results.

12.4.1 Model Description

The 2D vehicle model shown in Fig. 12.1 is used to validate the proposed estimation approach. It consists of a flexible vehicle body suspended on a spring-damper suspension system. The vehicle is loaded by a known gravitational force, the reaction forces due to the irregular road profile and an aerodynamic force. In total there are six unknown external forces that need to be estimated, as shown in Fig. 12.1. Table 12.1 summarizes the main properties of the vehicle body, Table 12.2 shows the properties of the spring-damper suspension system.

The system is modeled in the floating-frame-of-reference approach, and the reduction technique outlined in Sect. 12.2.1 is used to reduce the finite element model of the vehicle body. The residual inertia-relief attachment modes corresponding to the six force components together with the first six free-free normal modes are used to compute the reduced model, so that the number of flexible DOFs is equal to 12.

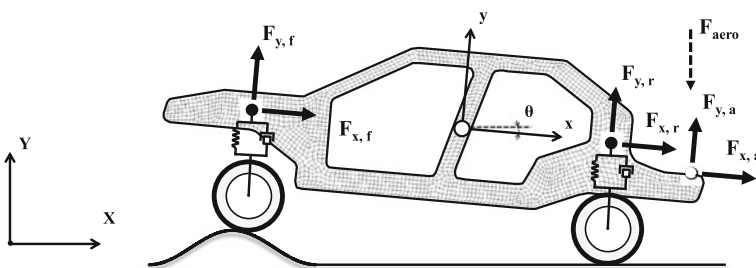


Fig. 12.1 2D vehicle model with unknown force components

Table 12.1 Main properties of the vehicle body

	Mass (kg)	Rot. iner. (kgm ²)	f ₁ (Hz)	f ₂ (Hz)	f ₃ (Hz)	nDOF (/)
Vehicle body	1000	1261	3.42	4.66	5.75	8726

Table 12.2 Properties of the suspension system

	Stiffness (N/m)	Damping (Ns/m)
Front	49e3	1e4
Rear	49e3	1e4

Eleven measurements are performed; ten strain measurements and one angular velocity measurement. The strain measurements are performed at high strain locations. The noise on the measurements is assumed to be Gaussian white noise with realistic values for the sensors. This system-measurement combination leads to a partially observable system. The unknown forces are observable, but the states \mathbf{q}_t , $\dot{\mathbf{q}}_t$ and θ are not. This is because no global position measurement is performed. The sampling frequency of the estimator is 1 kHz.

The measurements and the true force trajectories are generated using the reference model. The reference model is also derived according to the floating-frame-of-reference formulation. A high-accuracy Krylov reduced finite element model of the vehicle body is employed [25] of dimension 64. Convergence tests were performed to ascertain that this reduced model is a very good approximation of the full finite element model in a wide frequency range. The equations of motion for the reference model are integrated with a generalized- α solver with time step 1 ms and spectral radius $\rho_\infty = 1$ [26], to reduce the *inverse crime* [27].

The accuracy of the estimator model and the reference model is shown in Fig. 12.2 for a collocated measurement on a vertical and longitudinal load at the front wheel. From this figure it is clear that the reference model provides a very good reduced model in a wide frequency range. The accuracy of the estimator model is high for low frequencies, but decreases with increasing frequency. This is due to the nature of the residual inertia-relief attachment modes; they are constructed around 0 Hz.

The proposed approach, as described above, is compared to the current general approach where the vehicle body is assumed to be rigid in the model for the Kalman filter. In this case, three measurements are performed: an acceleration measurement, and an angular velocity measurement. The acceleration measurement provides accelerations in the frame attached to the vehicle body.

12.4.2 Simulation Results

For the numerical experiment, an imposed displacement is applied to both wheels and a concentrated aerodynamic force is applied at the rear of the vehicle. The imposed displacement is the superposition of a sine sweep (0–3 Hz) and a ramp. A time-delay

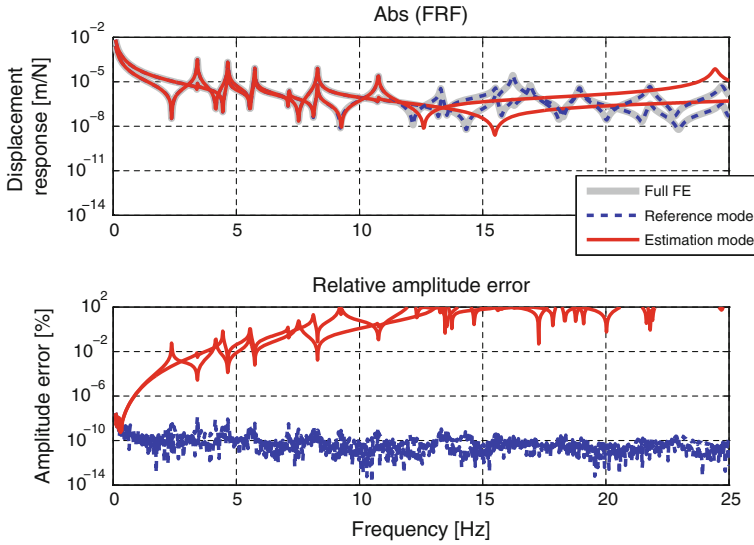


Fig. 12.2 Comparison of the full finite element model, reference model and estimator model. *Top*: magnitude of the frequency response function for a collocated measurement on a vertical and longitudinal load at the front wheel. *Bottom*: relative magnitude error in percent (%)

of 0.5 s exists between the excitation at the front and the rear wheel. The aerodynamic force is the superposition of a sine at constant frequency (3 Hz), and a constant load. In total there are six unknown external forces that need to be estimated, as shown in Fig. 12.1.

Figure 12.3 shows the simulation results. Part (a) of the figure clearly shows that the proposed approach is able to deliver accurate force estimates. At first sight, the accuracy of the estimate for $F_{x,a}$ seems rather low, but notice the low amplitude of this force as compared to the other forces. For the rigid approach the force estimates are significantly worse. This is not caused by the model-plant mismatch, but by the fact that in this case the system is unobservable. The three measurements do not provide sufficient information to differentiate between the six unknown force components. Performing any additional measurements, such as strain measurements, will not resolve this issue. This demonstrates the benefit of introducing flexible deformations in the model. It enables the observation of variables that cannot be obtained from a rigid model. This allows the filter to differentiate between the contributions of different input forces. Part (b) of Fig. 12.3 shows the evolution of the force estimate error variances. These results confirm the previous statements. The proposed approach leads to observability of the forces: the error variances quickly converge. The rigid approach leads to unobservability of the forces: the error variances diverge. Over time the estimates obtained with the rigid approach will further deteriorate.

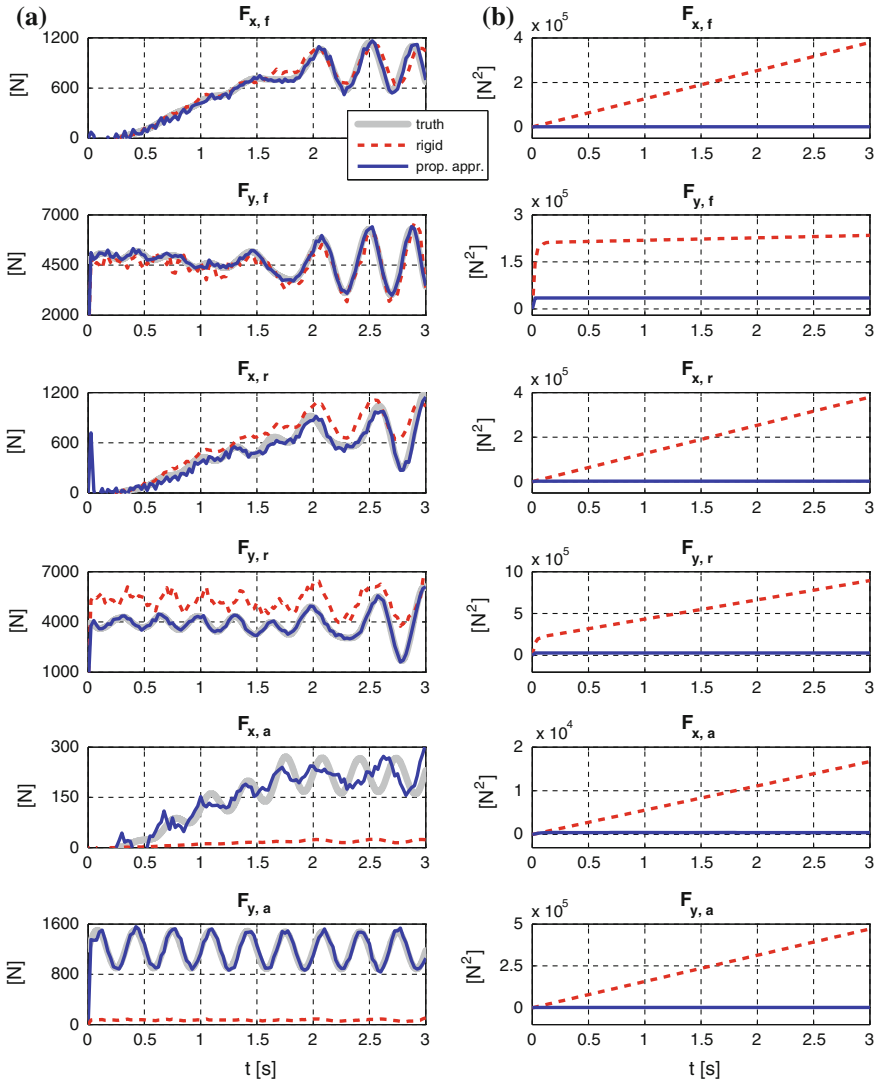


Fig. 12.3 Estimation results of the six unknown forces: **a** force estimates, **b** error variance of the force estimates

12.5 Conclusions

Accurate knowledge of wheel loads is of great value in vehicle design and control. However, a direct measurement of these forces is generally not feasible. This motivates the use of model-based estimation techniques such as the Kalman filter to leverage a limited set of measurements for more information. The general approach in literature is to use simple rigid models in the Kalman filter. This limits the possible estimation accuracy, and in many applications this results in an unobservable system, especially in the case of multiple unknown inputs. This paper proposed the use of general flexible multibody models for Kalman filtering. The introduction of flexible deformations in the model enables the observation of variables which cannot be obtained from a rigid model. This allows the filter to differentiate between the contributions of different input forces. This approach was demonstrated by employing the augmented extended Kalman filter to perform a combined estimation of the current vehicle state and wheel forces of a 2D vehicle model. The system was modeled in a floating-frame-of-reference approach and the vehicle body was described by a reduced order finite element model. It has been shown that, even though a global position measurement is required to obtain a fully observable system, strain measurements are sufficient to observe the unknown forces. The proposed approach was numerically validated and compared to an estimator with a rigid assumption. The validation demonstrated that the proposed approach provides superior estimation results. Future research will focus on extending the proposed methodology to spatial motion and multiple connected bodies.

Acknowledgments This work has been partially supported by the Flanders' DRIVE project 'GPS-positioning', granted by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). This work benefits from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO). The Fund for Scientific Research Flanders (F.W.O.), and the Research Fund KU Leuven are also gratefully acknowledged for their support.

References

1. Kalman RE (1960) A new approach to linear filtering and prediction problems. *J Basic Eng* 82(1):35–45
2. Simon D (2006) Optimal state estimation: Kalman, H_∞ and nonlinear approaches. Wiley, New York
3. Kitanidis PK (1987) Unbiased minimum-variance linear state estimation. *Automatica* 23(6):775–778
4. Darouach M, Zasadzinski M (1997) Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica* 33(4):717–719
5. Hostetter GH, Meditch JS (1973) On the generalization of observers to systems with unmeasurable, unknown inputs. *Automatica* 9(6):721–724
6. Ray LR (1995) Nonlinear state and tire force estimation for advanced vehicle control. *IEEE Trans Control Syst Technol* 3(1):117–124

7. Ray LR (1997) Nonlinear tire force estimation and road friction identification: simulation and experiments. *Automatica* 33(10):1819–1833
8. Doumiati M, Charara A, Victorino A, Lechner D (2012) Vehicle dynamics estimation using Kalman filtering. Wiley, New York
9. Wasfy T, Noor A (2002) Computational strategies for flexible multibody systems. *Appl Mech Rev* 56(6):553–613
10. Cuadrado J, Dopico D, Barreiro A, Delgado E (2009) Real-time state observers based on multibody models and the extended Kalman filter. *J Mech Sci Technol* 23(4):894–900
11. Cuadrado J, Dopico D, Perez JA, Pastorino R (2012) Automotive observers based on multibody models and the extended Kalman filter. *Multibody Syst Dyn* 27(1):3–19
12. Naets F, Pastorino R, Cuadrado J, Desmet W (2013) Online state and input force estimation for multibody models employing extended Kalman filtering. *Multibody Syst Dyn* 32(1):317–336
13. Naets F, Croes J, Desmet W (2015) An online coupled state/input/parameter estimation approach for structural dynamics. *Comput Methods Appl Mech Eng* 283:1167–1188
14. Naets F, Cuadrado J, Desmet W (2015) Stable force identification in structural dynamics using Kalman filtering and dummy-measurements. *Mech Syst Signal Process* 50–51:235–248
15. Lourens E, Reynders E, De Roeck G, Degrande G, Lombaert G (2012) An augmented Kalman filter for force identification in structural dynamics. *Mech Syst Signal Process* 27:446–460
16. Naets F, Heirman GHK, Desmet W (2011) Subsystem global modal parameterization for efficient simulation of flexible multibody systems. *Int J Numer Methods Eng* 89(1):1227–1248
17. Shabana AA (2005) *Dynamics Multibody Systems*, 3rd edn. Cambridge University Press, Cambridge
18. Zienkiewicz OC, Taylor RL (2000) *The finite element method*. Butterworth Heinemann, Oxford
19. Besselink B, Tabak U, Lutowska A, van de Wouw N, Nijmeijer H, Rixen DJ, Hochstenbach ME, Schilders WHA (2013) A comparison of model reduction techniques from structural dynamics, numerical mathematics and system and control. *J Sound Vib* 332(19):4403–4422
20. Craig RR, Kurdila AJ (2006) *Fundamentals of structural dynamics*, 2nd edn. Wiley, New York
21. Naets F, Pastorino R, Cuadrado J, Desmet W (2013) Online state and input force estimation for multibody models employing extended Kalman filtering. *Multibody Syst Dyn* 32(3):317–336
22. Ros J, Yoldi R, Plaza A, Angeles J (2012) Exponential integration schemes in multibody dynamics. In: *Proceedings of the second joint international conference on multibody dynamics*, Stuttgart, Germany
23. Van Loan CF (1978) Computing integrals involving the matrix exponential. *IEEE Trans Autom Control* 23(3):395–404
24. Ghosh BK, Rosenthal J (1995) A generalized Popov-Belevitch-Hautus test of observability. *IEEE Trans Autom Control* 40(1):176–180
25. Bai Z, Su Y (2005) Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method. *SIAM J Sci Comput* 26(5):1692–1709
26. Brüls O, Arnold M (2008) The generalized-alpha scheme as a linear multistep integrator: towards a general mechatronic simulator. *ASME J Comput Nonlinear Dyn* 3(4)
27. Kaipio J, Erkki E (2007) Statistical inverse problems: discretization, model reduction and inverse crimes. *J Comput Appl Math* 198(2):493–504

Chapter 13

Design and Control of an Energy-Saving Robot Using Storage Elements and Reaction Wheels

Makoto Iwamura, Shunichi Imafuku, Takahiro Kawamoto
and Werner Schiehlen

Abstract In this paper, we propose a concept for the design and control of an energy saving robot utilizing springs and reaction wheels. Firstly, we examine the simultaneous optimization problem of spring parameters and trajectories with respect to the energy consumption based on optimal control theory. We analyze the relationship between the consumed energy and the robot cycle time, derive a condition for an optimal energy efficient cycle, and propose a corresponding design method for springs. After that, we consider the practical design problem to realize the proposed energy saving manipulator concept. In order to verify the proposed method, a prototype 2DOF manipulator is developed by using linear springs and reaction wheels. The experimental results show the effectiveness of the proposed energy saving manipulator concept.

13.1 Introduction

In the manufacturing industry, machines and robots controlled by actuators are used to increase productivity and achieve high quality products. However, these actuators are consuming a great amount of energy accelerating and braking continuously. Hence, saving the energy of such mechanical systems is a very important issue.

Existing methods for reducing the energy consumption of industrial robots in manufacturing systems have been recently reviewed by Paryanto et al. [1]. There are three main approaches found as follows.

M. Iwamura (✉) · S. Imafuku · T. Kawamoto
Department of Mechanical Engineering, Fukuoka University,
Jonan, Fukuoka 814-0180, Japan
e-mail: iwamura@fukuoka-u.ac.jp

W. Schiehlen
Institute of Engineering and Computational Mechanics,
University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany
e-mail: schiehlen@itm.uni-stuttgart.de

- Energy-efficient motion planning:
 - Optimizing robot dynamics,
 - Collision free motion planning,
 - Smooth and time-optimal motion planning,
 - Optimal robot control systems,
 - Electrical energy exchange via internal DC bus.
- Optimal robot operation parameters:
 - Experimental approach,
 - Modelling and simulation approach,
 - Combination of experiments and simulation.
- Scheduling robot operations:
 - Optimal robot process sequences,
 - Start–stop optimization,
 - Optimizing robot cycle time.

Recuperated energy saving potential and approaches in industrial robotics were considered by Meike and Ribickis [2]. These authors present experimental and simulation results for two complementary alternatives: a capacitive energy buffer on the robot's DC-bus and a novel approach, the robot EnergyTeam. The principle of the robot EnergyTeam is a DC-bus merging among a various number of industrial robot power controllers thus enabling a controlled energy flow among the robots that decelerate and, therefore recuperate energy, and those that simultaneously require a peak power supply for the acceleration.

A path planning approach for the amplification of electrical energy exchange in multi axis robotic systems was presented by Hansen et al. [3]. The energy-based system model includes the robot dynamics, mechanical and drive losses, as well as the exchange of electrical energy. The nonlinear optimization problem is solved using global methods, considering kinematical and dynamic limitations. Simulations results are presented that prove the performance of the algorithm and demonstrate the beneficial effect of electrical energy exchange. Minimum time criteria can be retained if required and the approach is applicable to different multi axis manipulator types.

Resonant robotic systems were considered by Babitzky and Shipilov [4]. These robotic systems include spring accumulators and use control design with minimal energy consumption.

The standard task of industrial robots is a planned repetitive motion with a short cycle time. Such periodic motions result in phases of high accelerations and deceleration featuring high energy losses. Therefore, local energy storage transforming kinetic energy in reusable potential energy is attractive. Springs are reliable passive mechanical components for energy storage. Methods for reducing the consumed energy of controlled multibody systems by utilizing passive storage elements such as springs have been recently examined, see Refs. [5–7].

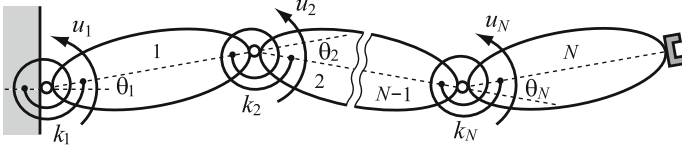


Fig. 13.1 Planar N DOF serial manipulator with storage springs

In this paper, we consider the case of a planar N DOF serial manipulator with springs as in shown Fig. 13.1. Firstly, we analyze theoretically how to choose the spring stiffnesses, spring mounting positions, and robot motion trajectories to maximize energy savings. Based on the results of the analysis, we propose a simultaneous optimization method of spring parameters and trajectories. We show the effectiveness of the proposed method by numerical simulations.

In theory, we can achieve zero energy consumption by our method, but in practice the method can not be directly applied to a conventional manipulator since it utilizes the free frictionless vibrations of the system. Therefore, we next consider the practical design and control problems to realize the proposed energy saving robot concept. We propose a design method that utilizes the linear springs and controlled reaction wheels and develop a prototype 2DOF energy saving manipulator based on the method. Finally, the effectiveness of the proposed method is verified through lab experiments.

13.2 Energy Saving Control Method Using Springs

This paper discusses the reduction of the energy consumption of SCARA robots by adding springs to the joints as shown in Fig. 13.1. In this chapter, we consider the simultaneous optimization problem of spring parameters and robot trajectories that minimizes the energy consumption based on the optimal control theory.

13.2.1 Problem Formulation

The equations of motion of N -link serial horizontal manipulators equipped with springs can be expressed as follows

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = -\mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\theta}_n) + \mathbf{u}, \quad (13.1)$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N] \in \mathbb{R}^N$ is the joint variable vector, $\mathbf{M} \in \mathbb{R}^{N \times N}$ is the inertia matrix, $\mathbf{h} \in \mathbb{R}^N$ is the vector of centrifugal and Coriolis forces, $\mathbf{u} = [u_1, u_2, \dots, u_N] \in \mathbb{R}^N$ is the vector of direct driving torques, $\mathbf{K} = \text{diag}[k_1, k_2, \dots, k_N] \in \mathbb{R}^{N \times N}$ is the rotational stiffness matrix, $\boldsymbol{\theta}_n = [\theta_{n1}, \theta_{n2}, \dots, \theta_{nN}] \in \mathbb{R}^N$ is the vector of

spring mounting positions which is vanishing in Fig. 13.1. Here, we consider motions between two points and rest at both ends, i.e.,

$$\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0, \quad \boldsymbol{\theta}(t_f) = \boldsymbol{\theta}_f, \quad (13.2)$$

$$\dot{\boldsymbol{\theta}}(0) = \mathbf{0}, \quad \dot{\boldsymbol{\theta}}(t_f) = \mathbf{0}, \quad (13.3)$$

where $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_f$ are the initial and final configurations and $t = t_f$ is the robot cycle time. The energy consumed by this motion can be estimated by the following cost function

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt, \quad f_0(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u}, \quad (13.4)$$

where $\mathbf{W} \in R^{N \times N}$ is a positive definite symmetric matrix.

The problem considered in this section can be formulated as follows: Find $\mathbf{u}(t)$, $\boldsymbol{\theta}(t)$, $\boldsymbol{\theta}_n$ and $\mathbf{k} = [k_1, k_2, \dots, k_N]^T$ by minimizing the energy consumption (13.4) subject to the initial and final conditions (13.2) and (13.3) for the system (13.1).

13.2.2 Theoretical Analysis of Minimum Energy Control

It is difficult to obtain analytical solutions of the problem formulated in Sect. 13.2.1 since the Eq. (13.1) has a strong nonlinearity. Hence, in this section, we analyze the problem approximately by using the linearization and modal analysis techniques.

Firstly, we shift the reference point to the middle point of the initial and final configurations in order to make the later calculations easy. We define $\boldsymbol{\theta}_m = \frac{1}{2}(\boldsymbol{\theta}_f + \boldsymbol{\theta}_0)$ and $\boldsymbol{\theta}_e = \frac{1}{2}(\boldsymbol{\theta}_f - \boldsymbol{\theta}_0)$, and shift the coordinates as $\tilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}(t) - \boldsymbol{\theta}_m$, $\tilde{\boldsymbol{\theta}}_n = \boldsymbol{\theta}_n - \boldsymbol{\theta}_m$. This transforms the initial and final conditions to the symmetric form

$$\tilde{\boldsymbol{\theta}}(0) = -\boldsymbol{\theta}_e, \quad \tilde{\boldsymbol{\theta}}(t_f) = \boldsymbol{\theta}_e, \quad (13.5)$$

$$\dot{\tilde{\boldsymbol{\theta}}}(0) = \mathbf{0}, \quad \dot{\tilde{\boldsymbol{\theta}}}(t_f) = \mathbf{0}. \quad (13.6)$$

And the equations of motion (13.1) is transformed to the following form

$$\tilde{\mathbf{M}}(\tilde{\boldsymbol{\theta}}) \ddot{\tilde{\boldsymbol{\theta}}} + \tilde{\mathbf{h}}(\tilde{\boldsymbol{\theta}}, \dot{\tilde{\boldsymbol{\theta}}}) + \mathbf{K} \tilde{\boldsymbol{\theta}} = \mathbf{u} + \mathbf{K} \tilde{\boldsymbol{\theta}}_n. \quad (13.7)$$

If it is assumed that fairly strong springs are used, the spring forces become predominant over the centrifugal and Coriolis forces. Therefore, we neglect these forces. Additionally, we assume that the inertia matrix can be approximated at the middle point $\tilde{\boldsymbol{\theta}} = \mathbf{0}$ ($\boldsymbol{\theta} = \boldsymbol{\theta}_m$) as $\mathbf{M}(\boldsymbol{\theta}_m) = \tilde{\mathbf{M}}(\mathbf{0}) \equiv \hat{\mathbf{M}} = \text{const}$. Under these assumptions, the following linearized equations of motion are obtained

$$\hat{\mathbf{M}} \ddot{\tilde{\boldsymbol{\theta}}} + \mathbf{K} \tilde{\boldsymbol{\theta}} = \mathbf{u} + \mathbf{K} \tilde{\boldsymbol{\theta}}_n. \quad (13.8)$$

Let us consider the free vibration system corresponding to Eq. (13.8) and calculate the modal matrix $\Phi \in R^{N \times N}$ that satisfies

$$\Phi^T \hat{M} \Phi = I, \quad \Phi^T K \Phi = \Omega^2, \quad \Omega = \text{diag}[\omega_1, \omega_2, \dots, \omega_N], \quad (13.9)$$

where I is the identity matrix, ω_i is the i th natural frequency arranged as $\omega_1 < \omega_2 < \dots < \omega_N$ where some of the lowest may vanish. We make the coordinate transformation $q = \Phi^{-1} \tilde{\theta}$ ($q_n = \Phi^{-1} \tilde{\theta}_n$) and define the state vectors $x = [x_1^T, x_2^T]^T = [q^T, \dot{q}^T]^T$. Then, the following state equations are found

$$\dot{x}_1 = x_2 \quad (\equiv f_1(x, u)), \quad (13.10)$$

$$\dot{x}_2 = -\Omega^2 x_1 + \Phi^T u + \Omega^2 q_n \quad (\equiv f_2(x, u)). \quad (13.11)$$

The initial and final conditions are expressed as

$$x_1(0) = -q_e, \quad x_1(t_f) = q_e, \quad (13.12)$$

$$x_2(0) = \mathbf{0}, \quad x_2(t_f) = \mathbf{0}, \quad (13.13)$$

where $q_e = \Phi^{-1} \theta_e$.

Next, let us introduce an adjoint vector $\psi = [\psi_1^T, \psi_2^T]^T$ and define the Hamiltonian as follows

$$H = f_0 + \psi_1^T f_1 + \psi_2^T f_2 = \frac{1}{2} u^T W u + \psi_1^T x_2 + \psi_2^T (-\Omega^2 x_1 + \Phi^T u + \Omega^2 q_n). \quad (13.14)$$

Then, the optimal control is derived from the condition $\partial H / \partial u = \mathbf{0}$ as

$$u = -W^{-1} \Phi \psi_2. \quad (13.15)$$

Substituting Eq. (13.15) into Eq. (13.14), the Hamiltonian along the optimal trajectory is given by

$$H = \psi_1^T x_2 - \psi_2^T \Omega^2 x_1 - \frac{1}{2} \psi_2^T \Phi^T W^{-1} \Phi \psi_2 + \psi_2^T \Omega^2 q_n. \quad (13.16)$$

From Eq. (13.16), the canonical equations of Hamilton can be derived as follows

$$\dot{x} = \partial H / \partial \psi = Ax + B\psi + c_n, \quad (13.17)$$

$$\dot{\psi} = -\partial H / \partial x = -A^T \psi, \quad (13.18)$$

where

$$A = \begin{bmatrix} \mathbf{0} & I \\ -\Omega^2 & \mathbf{0} \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\Phi^T W^{-1} \Phi \end{bmatrix}, \quad c_n = \begin{bmatrix} \mathbf{0} \\ \Omega^2 q_n \end{bmatrix}. \quad (13.19)$$

By solving the differential equations (13.17) and (13.18) under the boundary conditions (13.12) and (13.13), we obtain the optimal solution that minimizes the energy consumption. However, it is difficult to obtain the closed-form analytical solution of these equations since they are coupled due to the term $\Phi^T W^{-1} \Phi$ in the matrix B . In contrast, choosing the weighting matrix as $W = \hat{M}^{-1}$ results in the cost function of mechanical power and allows to decouple these equations by the property $\Phi^T W^{-1} \Phi = \Phi^T \hat{M} \Phi = I$. Therefore, in the following, we analyze this case.

Let us denote the initial value of adjoint vector as $\psi(0)$, then the solution of Eq. (13.18) reads

$$\psi(t) = e^{-A^T t} \psi(0). \quad (13.20)$$

By using Eq. (13.20), the solution of Eq. (13.17) is derived as

$$x(t) = e^{At} x(0) + e^{At} \int_0^t e^{-A\tau} B e^{-A^T \tau} d\tau \psi(0) + e^{At} \int_0^t e^{-A\tau} d\tau c_n. \quad (13.21)$$

If the final condition $x(t_f)$ is specified, $\psi(0)$ can be computed from Eq. (13.21) as follows

$$\psi(0) = \left[\int_0^{t_f} e^{-At} B e^{-A^T t} dt \right]^{-1} \times \left[e^{-At_f} x(t_f) - x(0) - \int_0^{t_f} e^{-At} dt c_n \right]. \quad (13.22)$$

Then, by substituting this $\psi(0)$ into Eq. (13.20), $\psi_2(t)$ is obtained as

$$\psi_2(t) = [\psi_{21}(t) \ \psi_{22}(t) \ \dots \ \psi_{2N}(t)]^T, \quad (13.23)$$

$$\psi_{2i}(t) = \frac{2\omega_i^2 \{\sin \omega_i(t_f - t) - \sin \omega_i t\}}{\sin \omega_i t_f - \omega_i t_f} q_{ei} + \frac{2\omega_i^2 \{\sin \omega_i(t_f - t) + \sin \omega_i t\}}{\sin \omega_i t_f + \omega_i t_f} q_{ni}, \quad (13.24)$$

where q_{ei} and q_{ni} is the i th element of q_e and q_n respectively. From Eq. (13.15), $W = \hat{M}^{-1}$, and Eq. (13.23), the optimal control $u(t)$ can be computed by

$$u(t) = -\hat{M} \Phi \psi_2(t). \quad (13.25)$$

Furthermore, by Eq. (13.25) and $W = \hat{M}^{-1}$, the cost function can be expressed as

$$\begin{aligned} J &= \frac{1}{2} \int_0^{t_f} u^T(t) \hat{M}^{-1} u(t) dt = \frac{1}{2} \int_0^{t_f} \psi_2^T(t) \psi_2(t) dt, \\ &= \sum_{i=1}^N \frac{2\omega_i^3 (1 + \cos \omega_i t_f)}{\omega_i t_f - \sin \omega_i t_f} q_{ei}^2 + \sum_{i=1}^N \frac{2\omega_i^3 (1 - \cos \omega_i t_f)}{\omega_i t_f + \sin \omega_i t_f} q_{ni}^2, \end{aligned} \quad (13.26)$$

where we use the fact that $\Phi^T \hat{M} \Phi = \mathbf{I}$. In the above equation, it is easily confirmed that

$$\frac{2\omega_i^3(1 - \cos \omega_i t_f)}{\omega_i t_f + \sin \omega_i t_f} \geq 0. \quad (13.27)$$

Hence it is understood that the optimal spring mounting position that minimizes J is always $q_{ni} = 0 (i = 1, 2, \dots, N)$, i.e., $\mathbf{q}_n = \mathbf{0} (\boldsymbol{\theta}_n = \boldsymbol{\theta}_m)$. Therefore in the following, we analyze the problem by setting $\mathbf{q}_n = \mathbf{0}$.

It should be noted that some of ω_i might be zero depending on the structure of the stiffness matrix \mathbf{K} . By using the L'Hospital's theorem, we can get

$$\lim_{\omega_i \rightarrow 0} \frac{2\omega_i^3(1 + \cos \omega_i t_f)}{\omega_i t_f - \sin \omega_i t_f} = \frac{24}{t_f^3}. \quad (13.28)$$

Hence, the relationship between the minimum value of the energy consumption J and the robot cycle time t_f including the case of $\omega_i = 0$ can be summarized as follows

$$J(t_f) = \sum_{i=1}^N J_i(t_f), \quad (13.29)$$

$$J_i(t_f) = \begin{cases} \frac{2\omega_i^3(1 + \cos \omega_i t_f)}{\omega_i t_f - \sin \omega_i t_f} q_{ei}^2 & (\text{if } \omega_i \neq 0) \\ \frac{24}{t_f^3} q_{ei}^2 & (\text{if } \omega_i = 0). \end{cases} \quad (13.30)$$

Equations (13.29) and (13.30) show that the total consumed energy of horizontal manipulators with springs can be reasonably understood as the sum of the consumed energy corresponding to the each mode.

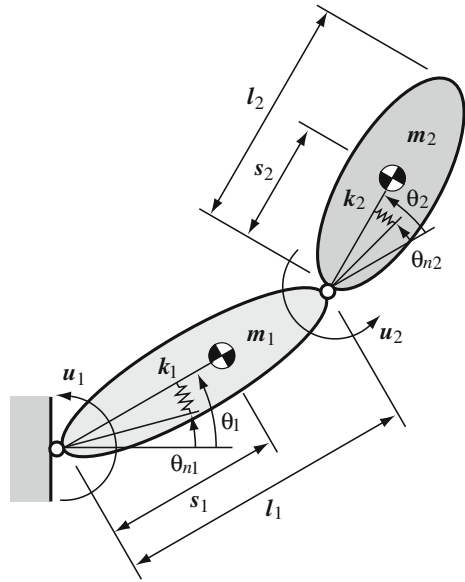
The optimal trajectory in modal coordinates $\mathbf{q}(t)$ is derived by substituting $\boldsymbol{\psi}(0)$, Eqs. (13.12) and (13.13) into Eq. (13.21) as follows

$$\mathbf{x}_1(t) = \mathbf{q}(t) = [q_1(t) \ q_2(t) \ \dots \ q_N(t)]^T, \quad (13.31)$$

$$q_i(t) = -q_{ei} \cos \omega_i t - q_{ei} \frac{(\omega_i t \cos \omega_i t - \sin \omega_i t)(1 + \cos \omega_i t_f)}{\sin \omega_i t_f - \omega_i t_f} - q_{ei} \frac{\omega_i t \sin \omega_i t \sin \omega_i t_f}{\sin \omega_i t_f - \omega_i t_f}. \quad (13.32)$$

Then, the optimal trajectory in physical coordinates $\boldsymbol{\theta}(t)$ can be obtained by $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_m + \Phi \mathbf{q}(t)$ where $\boldsymbol{\theta}_m$ is the middle position of the robot configuration as used for Eqs. (13.5) and (13.6).

Fig. 13.2 Planar 2DOF manipulator



13.2.3 Validation of the Obtained Optimal Solution

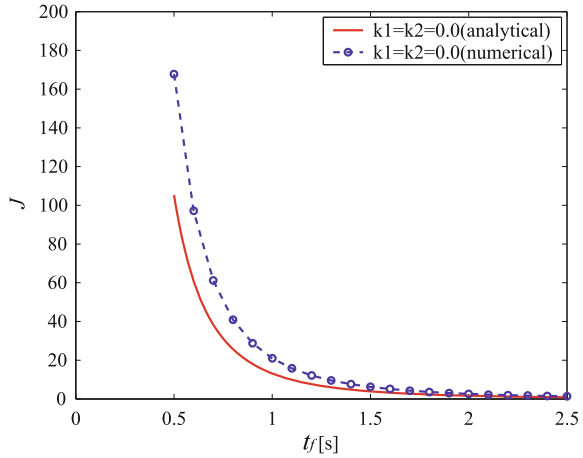
The analytical solution obtained in the previous section is based on the linearized equations of motion (13.8). Hence, we should examine the influence of the nonlinearity ignored in the analysis. Therefore, in this section, we compare the analytical solution with the numerical one considering the full nonlinear dynamics by a general purpose optimal trajectory planning algorithm for multibody systems [8]. Here we consider the planar 2DOF manipulator shown in Fig. 13.2 as an example. The parameters used for the simulation are given in Table 13.1.

The equations of motion of the planar 2DOF manipulator with springs can be expressed as

Table 13.1 Parameters of the 2DOF manipulator

Link	i	1	2
Length	l_i (m)	0.250	0.160
Center of mass	s_i (m)	0.125	0.080
Mass	m_i (kg)	14.25	10.00
Moment of inertia	I_i (kgm ²)	0.430	0.244

Fig. 13.3 Energy consumption J (without springs)



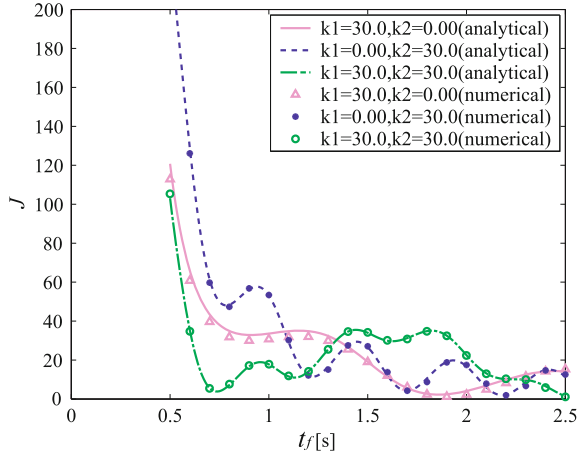
$$\begin{bmatrix} I_1 + I_2 + 2m_2l_1s_2 \cos \theta_2 + m_2(l_1^2 + s_2^2) + m_1s_1^2 & \text{sym.} \\ I_2 + m_2(s_2^2 + l_1s_2 \cos \theta_2) & I_2 + m_2s_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -m_2l_1s_2 \sin \theta_2 (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ m_2l_1s_2 \sin \theta_2 \dot{\theta}_1^2 \end{bmatrix} = - \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 - \theta_{n1} \\ \theta_2 - \theta_{n2} \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (13.33)$$

As an example, we consider the motion under the initial and final conditions $\theta_0 = [-30, -30]^T$ deg, $\theta_f = [30, 30]^T$ deg. Figure 13.3 shows the comparison between the analytical solution and the numerical one for the case without spring, i.e., $\mathbf{k} = [k_1, k_2]^T = [0, 0]^T$ Nm/rad. Obviously, short robot cycle times are very energy consuming. The centrifugal and Coriolis forces become larger as the moving speeds become faster, the difference between the two becomes larger as t_f becomes shorter. However, we can confirm that the difference converge to zero as t_f is longer, i.e., the moving speeds become slower. Figure 13.4 shows the comparison for the three cases of $\mathbf{k} = [k_1, k_2]^T = [30, 0]^T$, $[0, 30]^T$, $[30, 30]^T$ Nm/rad. If fairly strong springs are used, the spring forces become predominant over the centrifugal and Coriolis forces. Therefore, we can observe that both results are well coinciding everywhere. Since the analytical solution is well approximating the characteristics of the exact one, it is proposed that the analytical solution obtained in the previous section can be used for analysis and design of planar manipulators with springs. Furthermore, Fig. 13.4 indicates that for cycle time t_f less than 1 s the energy consumption may be considerably reduced by properly chosen springs.

13.2.4 Optimal Robot Cycle Time

In this subsection, we derive a condition for an optimal robot cycle time t_f . If some ω_i become zero, an optimal robot cycle time t_f does not exist since J has the terms

Fig. 13.4 Energy consumption J (reduced by proper springs)



that decrease monotonously with t_f , see Eq. (13.30). Therefore, in the following, we discuss the case that all ω_i are non-zero. If we consider t_f as the quantity that may take any value, $H(t_f) = 0$ should be satisfied from the transversality condition. Moreover, since Eq. (13.16) does not contain t explicitly, $\partial H/\partial t = 0$, it holds $H = \text{const}$ along optimal trajectories. Hence the condition $H(0) = 0$ should be satisfied. By substituting the Eqs. (13.12) and (13.13), $\mathbf{q}_n = \mathbf{0}$, and $\boldsymbol{\psi}(0)$ into Eq. (13.16), one can get

$$H(0) = \boldsymbol{\psi}_2^T(0)\boldsymbol{\Omega}^2\mathbf{q}_e - \frac{1}{2}\boldsymbol{\psi}_2^T(0)\boldsymbol{\psi}_2(0) = \sum_{i=1}^N \frac{-2\omega_i^5 t_f \sin \omega_i t_f}{(\omega_i t_f - \sin \omega_i t_f)^2} q_{ei}^2. \quad (13.34)$$

From Eq. (13.34), it follows that $H(0) = 0$ is satisfied if $\sin \omega_i t_f = 0$ ($i = 1, 2, \dots, N$) or equivalently

$$\omega_i t_f = r_i \pi \quad (i = 1, 2, \dots, N), \quad (13.35)$$

where r_i is an integer. When the condition (13.35) is satisfied, the second and third term of Eq. (13.24) vanishes and the optimal trajectory in modal coordinate reduce to harmonic vibrations

$$\mathbf{x}_1(t) = \mathbf{q}(t) = [q_1(t) \ q_2(t) \ \dots \ q_N(t)]^T, \quad (13.36)$$

$$q_i(t) = -q_{ei} \cos \omega_i t. \quad (13.37)$$

And by substituting Eq. (13.35) into Eqs. (13.29) and (13.30), the cost function can be expressed as

$$J = \sum_{i=1}^N J_i = \sum_{i=1}^N \frac{2\omega_i^3 (1 + \cos r_i \pi)}{r_i \pi} q_{ei}^2. \quad (13.38)$$

From Eq. (13.38), it is understood that J_i takes the maximum $4\omega_i^3 q_{ei}^2 / r_i \pi$ if r_i is an even number and vanishes if r_i is an odd number. Hence, J takes the global minimum if all r_i are odd number resulting in the minimum value of zero.

13.2.5 Optimal Design Method for Springs

In this section, we consider the problem to design the spring stiffnesses $\mathbf{k} = [k_1, k_2, \dots, k_N]^T$ that make the consumed energy minimum for a specified time t_f^* . Firstly, from Eq. (13.35), the natural frequencies ω_i read as

$$\omega_i = r_i \pi / t_f^* \quad (i = 1, 2, \dots, N), \quad (13.39)$$

where all r_i should be selected to be odd number so that all J_i takes the minimum. Moreover, r_i should satisfy $r_1 < r_2 < \dots < r_N$ since we assumed that $\omega_1 < \omega_2 < \dots < \omega_N$. The spring stiffnesses $\mathbf{k} = [k_1, k_2, \dots, k_N]^T$ should be determined as they satisfy the following characteristic equations

$$\det[\mathbf{K} - \omega_i^2 \hat{\mathbf{M}}] = 0 \quad (i = 1, 2, \dots, N). \quad (13.40)$$

Let us define the error vector $\mathbf{e} = [e_1, e_2, \dots, e_N]^T$ where $e_i = \det[\mathbf{K} - \omega_i^2 \hat{\mathbf{M}}]$. Then, the problem here becomes to find \mathbf{k} that satisfies

$$\mathbf{e}(\mathbf{k}) = \mathbf{0}. \quad (13.41)$$

Solving this nonlinear equation, e.g., by Newton–Raphson method, we can obtain the optimal spring stiffnesses \mathbf{k} that minimizes the energy consumption.

Especially, for the case of 2DOF manipulator, we can calculate the optimal spring stiffnesses analytically as follows. Since $N = 2$, from the characteristic equations $\det[\mathbf{K} - \omega_i^2 \hat{\mathbf{M}}] = 0$ ($i = 1, 2$), two equilateral hyperbolas can be obtained

$$k_2 = \frac{c_i^2}{k_1 - a_i} + b_i \quad (i = 1, 2), \quad (13.42)$$

where $a_i = \hat{m}_{11}(r_i \pi / t_f^*)^2$, $b_i = \hat{m}_{22}(r_i \pi / t_f^*)^2$, $c_i = \hat{m}_{12}(r_i \pi / t_f^*)^2$, and \hat{m}_{ij} is the ij th element of matrix $\hat{\mathbf{M}}$. From these two equations, k_1 can be calculated analytically as follows

$$k_1 = \frac{A^+ B^- - C^+ C^- \pm \sqrt{(C^+)^2 (C^-)^2 - 2A^- B^- (c_2^2 + c_1^2) + (A^-)^2 (B^-)^2}}{2B^-}, \quad (13.43)$$

where $A^+ = a_2 + a_1$, $A^- = a_2 - a_1$, $B^- = b_2 - b_1$, $C^+ = c_2 + c_1$, $C^- = c_2 - c_1$. Then, by substituting k_1 into Eq. (13.42), k_2 can also be obtained.

We can achieve the minimum energy control of planar robot manipulators by adding the springs with optimal stiffnesses to the joint at the optimal mounting positions θ_m .

13.2.6 Application Example

In this section, the proposed optimal design method for springs is demonstrated for the planar 2DOF manipulator shown in Fig. 13.2. As in the Sect. 13.2.3, we consider the motion under the initial and the final conditions $\theta_0 = [-30, -30]^T$ deg, $\theta_f = [30, 30]^T$ deg. Here we assume that the robot cycle time is specified as $t_f^* = 1$ s. Firstly, we select r_i in Eq. (13.39) as $r_1 = 1$, $r_2 = 3$, then the natural frequencies become $\omega_1 = \pi$, $\omega_2 = 3\pi$ rad/s. Let us first set the spring stiffnesses as $\mathbf{k} = [k_1, k_2]^T = [30, 30]^T$ Nm/rad. Though the consumed energy J corresponding to this spring stiffnesses is included in Fig. 13.4, we show it again on a different scale with J_1 and J_2 for the first and second mode in Fig. 13.5. Since the minimum of J_1 and J_2 do not coincide, J is not vanishing. Therefore, we optimize the spring stiffnesses. From Eqs. (13.42) and (13.43), the optimal values of spring stiffnesses can be obtained as $\mathbf{k} = [k_1, k_2]^T = [21.854, 14.182]^T$ Nm/rad. The consumed energy J corresponding to this spring stiffnesses is shown in Fig. 13.6 with J_1 and J_2 . We can observe that the minimum of J_1 and J_2 coincides at $t_f^* = 1$ s and therefore J is vanishing at $t_f^* = 1$ s. In Fig. 13.6, the minimum value of J corresponding to the case without spring, i.e., a conventional robot manipulator, is also shown. We can confirm that the consumed energy is strongly reduced at the design point $t_f^* = 1$ s. This proves the effectiveness of the proposed method.

Fig. 13.5 Energy consumption J before optimization

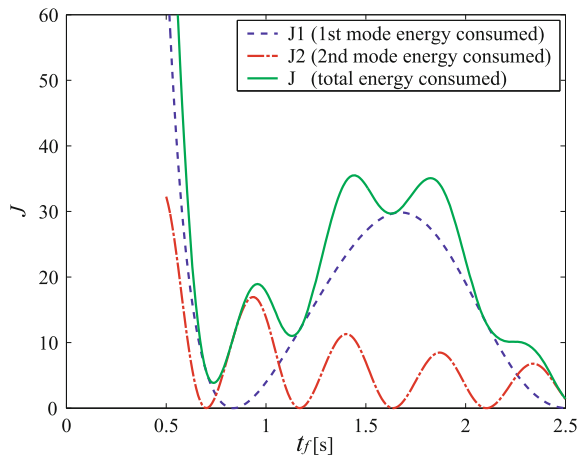
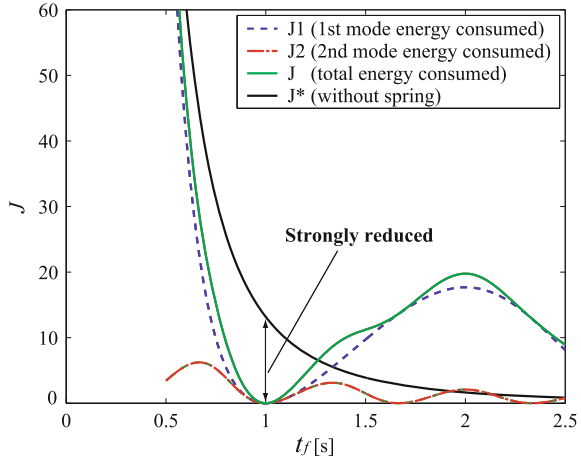


Fig. 13.6 Energy consumption J after optimization



13.3 Design and Control of Energy Saving Manipulator

The energy saving control method using springs proposed in the Chap. 2 utilizes natural modes of vibration of the system. On the other hand, existing robot manipulators have direct drive or geared motors at the joints and therefore free vibrations do not occur. Therefore, the proposed method cannot be directly applied to conventional manipulators. Hence in this chapter, we consider the practical design problem to realize the proposed energy saving manipulator concept. We develop a prototype 2DOF manipulator to validate realizability and effectiveness of the proposed method.

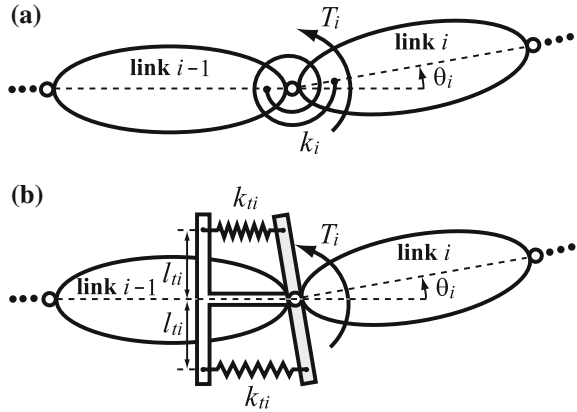
13.3.1 Design of Energy Saving Manipulator

In Chap. 2, we consider a horizontal manipulator model equipped with rotational springs as shown in Fig. 13.1 or Fig. 13.7a, respectively. However, for the rotational spring it is usually difficult to adjust its stiffness and mounting position, and so we impose in our experimental setup rotational stiffness between neighboring links by using two linear springs and a special spring holder as shown in Fig. 13.7b. We denote by k_{ii} the linear spring stiffness, l_{ii} the distance between the joint and spring mounting point on the holder, then the applied torque T_i by springs about a joint can be approximated as

$$T_i = -2k_{ii}l_{ii}^2 \sin \theta_i \cos \theta_i \simeq -2k_{ii}l_{ii}^2 \theta_i (\equiv -k_i \theta_i). \tag{13.44}$$

From the above equation, it is understood that we can convert linear spring stiffness k_{ii} to corresponding rotational spring stiffness k_i by

Fig. 13.7 Structure of proposed energy saving robot, **a** mathematical model, **b** practical design



$$k_i = 2k_{ti}l_{ti}^2. \tag{13.45}$$

Next, let us consider the installation position of actuators. Since the proposed method utilizes the free vibration of the system, all joints must be able to rotate freely. Therefore, we will not install motors at the joints, instead, we introduce controlled reaction wheels at an arbitrary point on the link and add driving torques from them as shown in Fig. 13.8.

The equations of motion of the links and the reaction wheels in Fig. 13.8 can be derived as follows

$$\mathbf{M}_{\theta\theta}\ddot{\boldsymbol{\theta}} + \mathbf{M}_{\phi\theta}^T\ddot{\boldsymbol{\phi}} + \mathbf{h} = -\mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\theta}_n), \tag{13.46}$$

$$\mathbf{M}_{\phi\theta}\ddot{\boldsymbol{\theta}} + \mathbf{M}_{\phi\phi}\ddot{\boldsymbol{\phi}} = \boldsymbol{\tau}, \tag{13.47}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]^T$ is the vector of joint variables, $\boldsymbol{\phi} = [\phi_1, \phi_2, \dots, \phi_N]^T$ is the vector of rotation angles of reaction wheels, $\mathbf{M}_{\theta\theta}$, $\mathbf{M}_{\phi\theta}$, $\mathbf{M}_{\phi\phi}$ are the inertia matrices, \mathbf{h} is the vector of centrifugal and Coriolis forces, $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_N]^T$

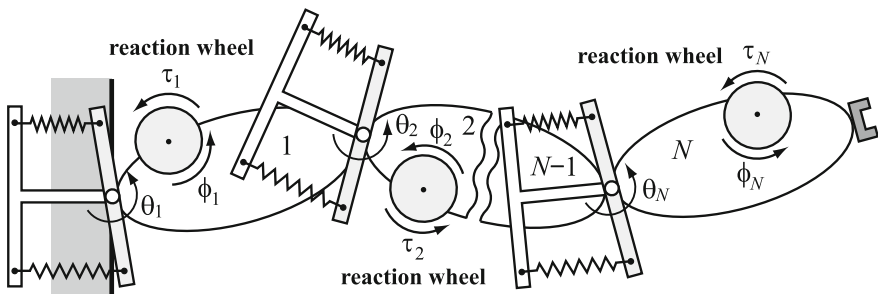


Fig. 13.8 Structure of proposed energy saving robot

is the vector of driving torques of reaction wheels, $\mathbf{K} = \text{diag}[k_1, k_2, \dots, k_N]$ is the stiffness matrix, $\boldsymbol{\theta}_n$ is the vector of spring mounting positions.

By eliminating $\boldsymbol{\phi}$ from Eqs.(13.46) and (13.47), and defining $\mathbf{M} \equiv \mathbf{M}_{\theta\theta} - \mathbf{M}_{\phi\theta}^T \mathbf{M}_{\phi\phi}^{-1} \mathbf{M}_{\phi\theta}$, $\mathbf{u} \equiv -\mathbf{M}_{\phi\theta}^T \mathbf{M}_{\phi\phi}^{-1} \boldsymbol{\tau}$, then the following equations of motion are obtained

$$\mathbf{M}\ddot{\boldsymbol{\theta}} + \mathbf{h} = -\mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\theta}_n) + \mathbf{u}. \quad (13.48)$$

Since Eq.(13.48) has the same form as Eq.(13.1), the energy saving control method proposed in the Chap. 2 can be directly applied to this novel energy saving manipulator systems.

13.3.2 Control of Energy Saving Manipulator

From Eqs.(13.36) and (13.37), the minimum energy trajectory between the initial position $\boldsymbol{\theta}_0 = [\theta_{10}, \theta_{20}, \dots, \theta_{N0}]^T$ and the final position $\boldsymbol{\theta}_f = [\theta_{1f}, \theta_{2f}, \dots, \theta_{Nf}]^T$ can be expressed as

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}_m + \boldsymbol{\Phi}\mathbf{q}(t) (\equiv \boldsymbol{\theta}_d(t)), \quad (13.49)$$

$$\mathbf{q}(t) = [-q_{e1} \cos \omega_1 t, -q_{e2} \cos \omega_2 t, \dots, -q_{eN} \cos \omega_N t]^T. \quad (13.50)$$

Theoretically, if we move the links to the position $\boldsymbol{\theta}_0$ by applying the external torques for the first time only, then repetitive movement between $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_f$ can be achieved continuously without any additional input torque \mathbf{u} . However, practically, due to the existence of friction and air resistance, the link motions attenuate gradually with time. Hence, let the desired trajectory $\boldsymbol{\theta}_d(t)$ be given by Eq.(13.49), and following feedback controller is introduced

$$\mathbf{u} = \mathbf{M}\{\ddot{\boldsymbol{\theta}}_d - \boldsymbol{\alpha}(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_d) - \boldsymbol{\beta}(\boldsymbol{\theta} - \boldsymbol{\theta}_d)\} + \mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\theta}_n). \quad (13.51)$$

By substituting Eq.(13.51) into the equations of motion (13.48), it can be seen that the behavior of the error vector $\mathbf{e}(t) \equiv \boldsymbol{\theta}(t) - \boldsymbol{\theta}_d(t)$ is governed by the equation

$$\ddot{\mathbf{e}}(t) + \boldsymbol{\alpha}\dot{\mathbf{e}}(t) + \boldsymbol{\beta}\mathbf{e}(t) = \mathbf{0}, \quad (13.52)$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are constant matrices that guarantee asymptotic stability. Equation(13.52) means that $\boldsymbol{\theta}(t)$ converges to $\boldsymbol{\theta}_d(t)$, which means that proposed energy saving control method is realized.

Fig. 13.9 *Top view of the prototype 2DOF manipulator*

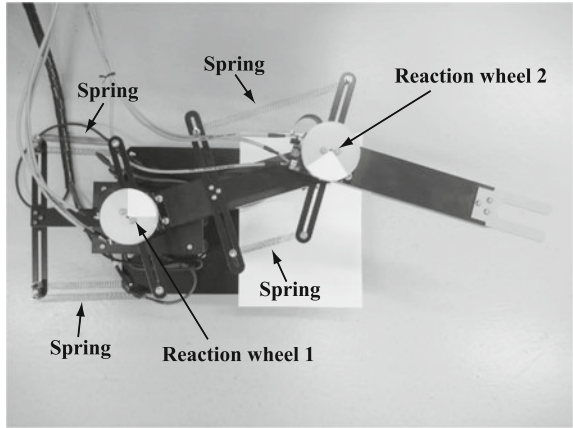
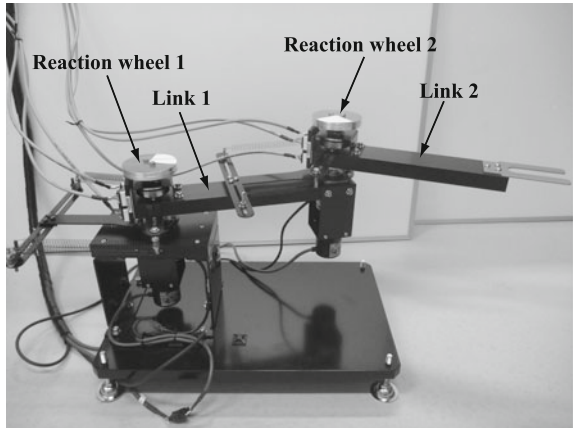


Fig. 13.10 *Side view of the prototype 2DOF manipulator*



13.3.3 A Prototype 2DOF Manipulator and Experimental Results

To validate the proposed energy saving control method, and practical design and control approach, we develop a prototype 2DOF manipulator. The developed energy saving manipulator is shown in Figs. 13.9 and 13.10. The main specifications of the experimental setup are in Tables 13.2, 13.3 and 13.4. In this experimental device, reaction wheels are driven by DC motors and installed such that their rotational center coincide with the joint axes. The rotary encoders are used to obtain rotational angles of the links. And the electro-magnetic breaks are equipped to the joints so that we can keep holding the links at an arbitrary positions.

Table 13.2 Parameters of the 2DOF energy saving manipulator

Link	i	1	2
Length	l_i (m)	0.2500	0.2250
Mass	m_i (kg)	0.8097	0.6747
Moment of inertia	I_i (kgm ²)	0.0062	0.0037

Table 13.3 Parameters of the reaction wheel

Reaction wheel	i	1	2
Mass	m_{wi} (kg)	0.0484	0.0484
Moment of inertia	I_{wi} (kgm ²)	2.767×10^{-5}	2.767×10^{-5}

Table 13.4 Parameters of the motor

Motor	i	1	2
Mass	m_{di} (kg)	0.0750	0.0750
Moment of inertia	I_{di} (kgm ²)	9.250×10^{-6}	9.250×10^{-6}

As an example, let us consider the repetitive movement between $\theta_0 = [0, 0]^T$ deg and $\theta_f = [60, 60]^T$ deg. The robot cycle time between two points are set as $t_f^* = 2$ s. If we choose r_i as $r_1 = 1$, $r_2 = 3$, the optimal rotational spring stiffnesses can be obtained as $\mathbf{k} = [k_1, k_2]^T = [0.374, 0.085]^T$ Nm/rad from Eqs. (13.42) and (13.43). By using Eq. (13.45), \mathbf{k} is converted to the linear spring stiffnesses as $\mathbf{k}_t = [k_{t1}, k_{t2}]^T = [18.706, 4.237]^T$ N/m. Then we adjust the linear spring stiffnesses of the experimental device as close as possible to the optimal values. After that we conduct trajectory tracking control experiments by using the control law of Eq. (13.51). Figures 13.11 and 13.12 show the results of the trajectory tracking test for the 1st and 2nd joint angles respectively. In these figures, dotted line shows the optimal (reference) trajectory calculated by Eqs. (13.49) and (13.50), and solid line shows the actual trajectory measured by encoders. From these figures, it can be seen that the tracking error is fairly small and trajectory tracking is almost achieved. Figures 13.13 and 13.14 show the driving torques of 1st and 2nd actuators respectively. In these figures, the solid line shows the reaction wheel driving torque of proposed energy saving manipulator and dotted line shows the joint driving torque required by the conventional manipulator to perform the same task. Let us compare the energy consumption of the proposed and the conventional manipulators by the following cost function

$$J(t) = \frac{1}{2} \sum_{i=1}^2 \int_0^t u_i^2(\tilde{t}) d\tilde{t}. \quad (13.53)$$

Fig. 13.11 Angle versus time (joint 1)

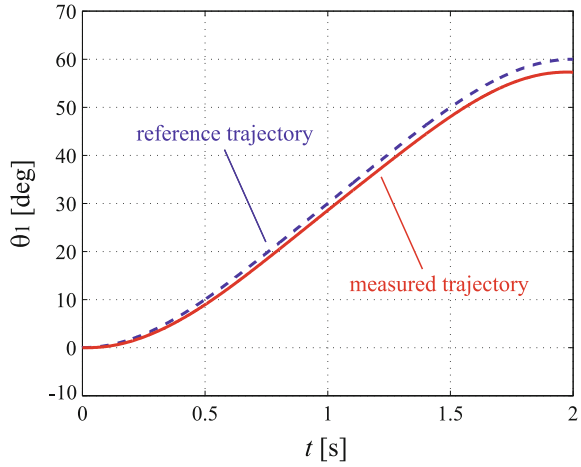


Fig. 13.12 Angle versus time (joint 2)

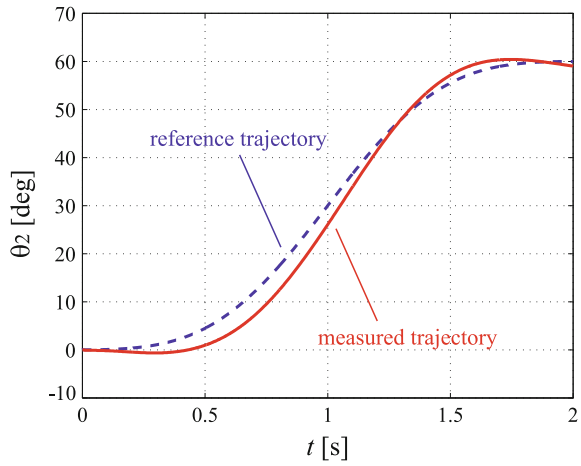


Figure 13.15 shows the result. The value of $J(t_f)$ for the proposed manipulator was 10.57, in contrast for the conventional manipulator was 173.5. It turns out that the energy consumption is reduced by 94% in the experiment. In theory, see Sect. 13.2.6, we could show 100% reduction by optimization but 94% is still an excellent number. This result proves that the proposed concept is effective and it can strongly reduce the energy consumption.

Fig. 13.13 Torque versus time (actuator 1)

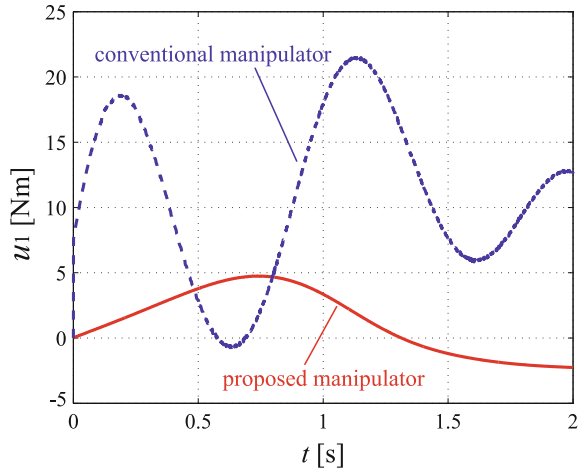


Fig. 13.14 Torque versus time (actuator 2)

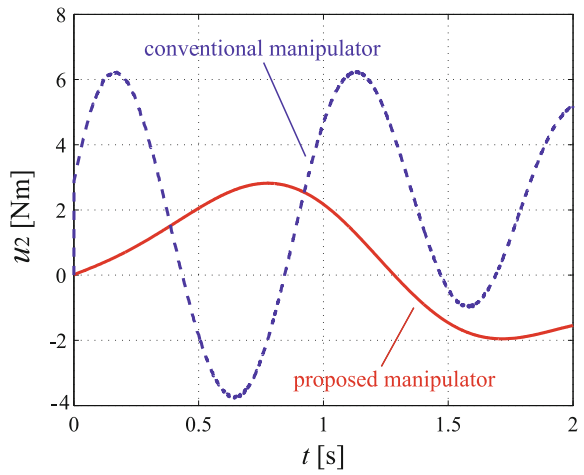
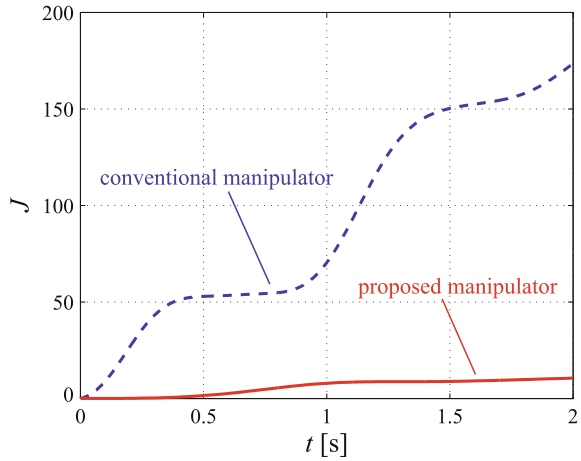


Fig. 13.15 Comparison of energy consumption



13.4 Conclusions

In this study, we considered a method for reducing energy consumption of planar robot manipulators by adding springs to the joints and utilizes the potential energy effectively. Firstly, we proposed a simultaneous optimization method for springs and trajectories based on the optimal control theory. Then, we discussed the practical design and control problems to realize the proposed energy saving manipulator concept. Finally, we developed a 2DOF energy saving manipulator prototype and show the effectiveness of the proposed method through experiments.

Acknowledgments This research was supported by KAKENHI, Grants-in-Aid for Young Scientists (B) (No. 26820076).

References

1. Paryanto P, Brossog M, Bornschlegl M, Franke J (2015) Reducing the energy consumption of industrial robots in manufacturing systems. *Int J Adv Manuf Technol* 78(5):1315–1328
2. Meike D, Ribickis L (2011) Recuperated energy savings potential and approaches in industrial robotics. In: 2011 IEEE conference on automation science and engineering (CASE), pp 299–303. doi:[10.1109/CASE.2011.6042435](https://doi.org/10.1109/CASE.2011.6042435)
3. Hansen C, Kotlarski J, Ortmaier T (2013) Path planning approach for the amplification of electrical energy exchange in multi axis robotic systems. In: 2013 IEEE International conference on mechatronics and automation (ICMA). doi:[10.1109/ICMA.2013.6617891](https://doi.org/10.1109/ICMA.2013.6617891)
4. Babitzky VI, Shipilov AV (2003) Resonant robotic systems. Springer, Berlin

5. Schiehlen W, Guse N (2005) Powersaving control of mechanisms. In: Ulbrich H, Gunthner W (eds) *Vibration control of nonlinear mechanisms and structures*. Springer, Dordrecht, pp 277–286
6. Schiehlen W, Iwamura M (2009) Minimum energy control of multibody systems utilizing storage elements, In: *Proceedings of the ASME 2009 international design engineering technical conferences*, 86327
7. Iwamura M, Schiehlen W (2011) Minimum control energy in multibody systems using gravity and springs. *J Syst Design Dyn* 5(3):474–485
8. Iwamura M, Eberhard P, Schiehlen W, Seifried R (2009) A general purpose optimal trajectory planning algorithm for multibody systems. In: *Proceedings of the 28th IASTED international conference on modeling, identification, and control*, pp 55–62

Chapter 14

Exploiting the Equations of Motion For Biped Robot Control with Enhanced Stability

Johannes Mayr, Alexander Reiter, Hubert Gattringer
and Andreas Müller

Abstract The scope of the present contribution is the derivation of the equations of motion and its field of application for humanoid robots, in particular legged robots. The derivation is performed in a modular and structured manner and it is shown how these equations can be exploited for the control of biped robots. The used methods allow to easily adopt the kinematic structure of single limbs and to reuse results obtained for limbs with similar kinematic structure but different inertial parameters such as in case the left leg is a mirrored version of the right one. After finding a recursive formulation to calculate the equations of motion we perform various state transformations and apply some model simplifications to obtain expressions that can be used to efficiently solve control problems. Two applications, compensating for the overall angular momentum and calculation of feed-forward torques, are shown. In both applications we can exploit the recursive calculation of the equations of motion used during the subsystem synthesis giving rise to real-time algorithms that can be used on a physical humanoid robot system.

Keywords Multi-body modeling · Angular momentum · Inverse dynamics · Biped control

J. Mayr (✉) · A. Reiter · H. Gattringer · A. Müller
Institute of Robotics, Johannes Kepler University Linz,
Linz, Austria
e-mail: joh.mayr@jku.at

A. Reiter
e-mail: alexander.reiter@jku.at

H. Gattringer
e-mail: hubert.gattringer@jku.at

A. Müller
e-mail: a.mueller@jku.at

14.1 Introduction

For the derivation of the equations of a complex multi-body model various different algorithms can be found in the literature. Due to computational efforts in robotics typically recursive algorithms like [10, 17] are used to calculate the direct dynamics. For the calculation of generalized accelerations algorithms like [8, 16] can be used. In [1, 9] an algorithm based on subsystems is introduced. This algorithm allows to save up calculation time by exploiting the similarities within the kinematic structure of robots. Both, the direct dynamics and the generalized accelerations can be calculated in a recursive way using these subsystems [1, 9].

When stabilizing a biped humanoid robot system one has to cope with many different control challenges mainly introduced by the high amount of degrees of freedom and the under-actuation of the system. To allow the robot to walk stably without tripping over one has to consider the contact forces between the robot and the environment. Typically the stabilization of the robot and the generation of the walking pattern are treated separately [3, 6, 12, 19] which allows to use models of different complexity for the different tasks. First a set of trajectories is defined according to the desired walking behavior [3, 5, 12, 18]. This trajectory is then tracked as accurately as possible. In the ideal case the robot would be able to walk dynamically balanced just by tracking those trajectories, but due to the unknown environment, inaccurate parameters and external disturbances different control layers have to modify the generated walking pattern to stabilize the robot around the given equilibrium trajectory. Especially the walking pattern generation relies on an accurate system model given by the equations of motion for the multi-body system [5, 18]. Depending on the desired walking behavior (e.g. fast walking) more accurate models are needed for walking pattern generator to provide physically consistent trajectories. On the other hand, complex models are computationally heavy and algorithms based on them may not be executed in realtime. Thus one needs to find a reasonable trade-off between model accuracy and computational effort of the model. Various different models have been proposed in the literature. The simplest one is given by a single pointmass at constant height and is typically referred to as three dimensional linear inverted pendulum mode (3D-LIPM) [14]. An extension of this simple model, which adds two other masses for the legs to the model, is used in [3]. Various offline walking pattern generators use the full or a slightly reduced multi-body model of the robot [5, 18].

In this communication we are going to present various different models that can be used for the control of a biped robot. We start with the equations of motion for the full multi-body model and after various simplifications we will end up with the three dimensional linear inverted pendulum. Exemplarily two control applications, the compensation of the angular momentum of the robot while walking, and the calculation of feed-forward torques will demonstrate how the different models can be used. The compensation of the angular momentum of the robot is already shown in [13] and realized by a resolved momentum controller. In this work we used a slightly different approach to obtain the Jacobian for the angular momentum directly from

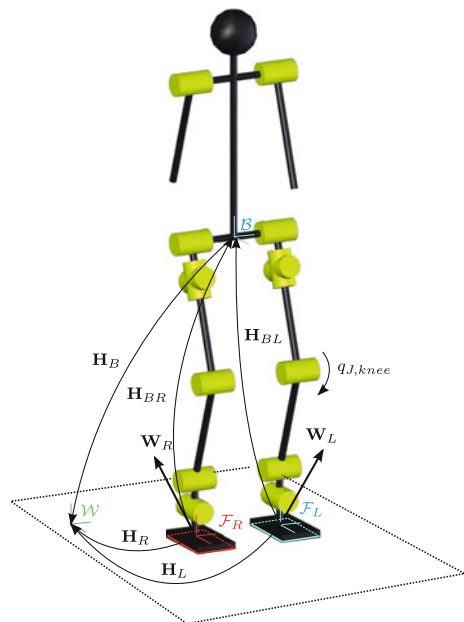
the equations of motion in minimal representation. The calculation of feed-forward torques for the quasi-static case is shown in [11]. Therein a least-square solution is used to calculate contact forces and joint torques for the current gravity forces. In [21] an extended version is presented that solves a quadratic program to calculate the optimal contact forces. These contact forces can then be used to calculate the feed forward torques for the joints.

The text is organized as follows. After a short introduction to the used multi-body algorithm a detailed multi-body model and various simplified models of the robot will be derived in Sects. 14.2 and 14.3. In Sect. 14.4 these models will then be used to exemplarily show two possible control applications. Section 14.5 concludes the contribution.

14.1.1 Coordinates and Frames

The frames and their origin used during the following sections are summarized in Fig. 14.1. The frame \mathcal{W} is a world fixed frame with the z -axis pointing in the opposite direction of the gravity vector \mathbf{g} . A body fixed frame \mathcal{B} is attached to the torso. For the description of the position of the limbs the frames $\mathcal{F}_R, \mathcal{F}_L$ are fixed to the left leg and the right leg. While the relative rotation and translation of the origins between two frames \mathcal{B} and \mathcal{L} is given by the homogenous transformation $\mathbf{H}_{BL} \in \text{SE}(3)$, the position and orientation of a frame \mathcal{B} with respect to the world frame \mathcal{W} is given by

Fig. 14.1 Frames and variables used for the kinematic description



$\mathbf{H}_B \in \text{SE}(3)$.¹ The twist between two frames is given by $\mathbf{V}_{BL}^b \in \mathbb{R}^6$ and corresponds to the linear and angular velocity of the frame \mathcal{L} relative to frame \mathcal{B} represented in \mathcal{L} . The superscript $\{\}^b$ denotes a body twist. The twist of frame \mathcal{B} with respect to \mathcal{W} is simply given by $\mathbf{V}_B^b \in \mathbb{R}^6$.

14.1.2 Configuration Space

The kinematic configuration of robotic manipulators is typically described by their joint positions. For a humanoid robot this is not sufficient. Depending on the contact state of the robot additional information is needed to fully define the position of the robot in the world frame. In the worst case, there is no contact of the robot to the ground. This state will be referred to as free floating base. Adding the position and orientation of a frame attached to an arbitrary link (e.g. \mathcal{B} for the base frame) to the configuration space leads to the complete kinematic configuration of a free floating robot. Thus the configuration space is given by

$$\mathcal{Q} = \{(\mathbf{H}_B, \mathbf{q}_J) | \mathbf{H}_B \in \text{SE}(3), \mathbf{q}_J \in \mathcal{J}\} \quad (14.1)$$

with $\mathcal{J} = \mathcal{J}_1 \times \dots \times \mathcal{J}_n$ as the joint space of a robot with n drive units and $\mathcal{J}_i \subseteq \mathbb{R}^m$ as the parameter space representing the m degrees of freedom of the i th drive unit. Further \mathbf{H}_B is the homogenous transformation from a world fixed frame to the upper-body, \mathbf{q}_J summarizes all joint positions.

The generalized velocities corresponding to the configuration space \mathcal{Q} are chosen as

$$\dot{\mathbf{s}} = \begin{bmatrix} \mathbf{V}_B^b \\ \dot{\mathbf{q}}_J \end{bmatrix}. \quad (14.2)$$

Note that $\dot{\mathbf{s}}$ contains non-holonomic velocities.

14.1.3 Task Space

To simplify trajectory and controller design it is useful to define different task spaces for the robot:

$$\mathcal{T}_1 = \{(\mathbf{r}_c, \mathbf{R}_B, \mathbf{q}_J) | \mathbf{r}_c \in \mathbb{R}^3, \mathbf{R}_B \in \text{SO}(3), \mathbf{q}_J \in \mathcal{J}\} \quad (14.3)$$

and

$$\mathcal{T}_2 = \{(\mathbf{r}_c, \mathbf{R}_B, \mathbf{H}_R, \mathbf{H}_L, \hat{\mathbf{q}}_J) | \mathbf{r}_c \in \mathbb{R}^3, \mathbf{R}_B \in \text{SO}(3), \mathbf{H}_i \in \text{SE}(3), \hat{\mathbf{q}}_J \in \hat{\mathcal{J}}\} \quad (14.4)$$

¹For simplification, the index of the world fixed frame \mathcal{W} is suppressed.

with $\hat{\mathcal{J}} \subset \mathcal{J}$. In contrast to \mathcal{Q} , for \mathcal{T}_1 the inertial center of mass (CoM) position \mathbf{r}_c is used to describe the position of the robot in space. The orientation of the upper-body is given by the rotation matrix \mathbf{R}_B . In \mathcal{T}_2 the joint positions of the legs are replaced by the homogeneous transformations to the feet. The joint positions of the arms $\hat{\mathbf{q}}_J$ remain in the task space.

The corresponding velocities are chosen to

$$\dot{\mathbf{z}}_1 = [\dot{\mathbf{r}}_c^T \ \omega_B^T \ \dot{\mathbf{q}}_J^T]^T \quad (14.5)$$

and

$$\dot{\mathbf{z}}_2 = [\dot{\mathbf{r}}_c^T \ \omega_B^T \ (\mathbf{V}_R^b)^T \ (\mathbf{V}_L^b)^T \ \dot{\mathbf{q}}_J^T]^T \quad (14.6)$$

with angular velocity of the upper body ω_B .

14.2 Dynamic Modeling

14.2.1 Subsystem Modeling

In general the equations of motion for a robot with n degrees of freedom and a free floating base is given in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{s}} + \mathbf{G}(\mathbf{q}, \dot{\mathbf{s}})\dot{\mathbf{s}} - \mathbf{Q}(\mathbf{q}) = 0 \quad (14.7)$$

where $\mathbf{q} \in \mathcal{Q}$ is the tuple of generalized coordinates. The generalized non-holonomic velocities $\dot{\mathbf{s}} \in \mathbb{R}^{6+n}$ are given by (14.2). $\mathbf{M} \in \mathbb{R}^{(6+n) \times (6+n)}$ denotes the mass matrix, $\mathbf{G} \in \mathbb{R}^{(6+n) \times (6+n)}$ is the matrix of the centrifugal and Coriolis forces and $\mathbf{Q} \in \mathbb{R}^{6+n}$ is the vector of generalized forces and torques. In this work an approach based on the Projection Equation introduced in [2] is used to derive the multi-body dynamics from (14.7). For a multi-body system with N bodies the equations of motion are given by

$$\sum_{i=1}^N \left[\left(\frac{\partial {}_R \mathbf{v}_c}{\partial \dot{\mathbf{s}}} \right)^T \left(\frac{\partial {}_R \boldsymbol{\omega}}{\partial \dot{\mathbf{s}}} \right)^T \right]_i \left[{}_R \dot{\mathbf{p}} + {}_R \tilde{\boldsymbol{\omega}}_{IR} \mathbf{R} \mathbf{p} - {}_R \mathbf{f}^e \right]_i = 0. \quad (14.8)$$

with ${}_R \mathbf{p}_i = m_i \mathbf{v}_{c,i}$ being the vector of linear momentum and ${}_R \mathbf{L}_i = {}_R \mathbf{J}_{c,i} {}_R \boldsymbol{\omega}_i$ the vector of angular momentum. The absolute velocity of the CoM and the angular velocity of the i th body are given by ${}_R \mathbf{v}_{c,i} \in \mathbb{R}^3$ and ${}_R \boldsymbol{\omega}_i \in \mathbb{R}^3$. The inertia tensor of the i th body is ${}_R \mathbf{J}_{c,i} \in \mathbb{R}^{3 \times 3}$. The angular velocity of the reference frame for the i th body is given by ${}_R \boldsymbol{\omega}_{IR,i} \in \mathbb{R}^3$. External forces and torques like motor torques and gravitational forces acting on the CoM of the i th body are considered by ${}_R \mathbf{f}_i^e \in \mathbb{R}^3$ and ${}_R \mathbf{M}_i^e \in \mathbb{R}^3$. The $\tilde{(\cdot)}$ operator describes the skew symmetric matrix for calculating the cross product ($\tilde{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$, $\tilde{\mathbf{a}} \in \text{SO}(3)$). For each body i in (14.8) a different arbitrary reference frame R can be used. Unlike other algorithms the Projection

Equation allows the use of non-holonomic velocities and non-holonomic constraints a priori. Thus, the angular velocity of the base can directly be used as part of the generalized velocities. With the Projection Equation it is even possible to describe multi-body systems consisting of a combination of rigid and elastic bodies within the same framework. For further explanation see [2].

For the proposed system and for robotic systems in general, it is useful to combine interconnected bodies to subsystem assemblies (e.g. motor—gear—structural components). This can be done by partitioning the sum of (14.8) into N_s subsystems

$$\sum_{i=1}^{N_s} \left\{ \sum_{j=1}^{N_i} \left[\left(\frac{\partial_R \mathbf{v}_c}{\partial \dot{\mathbf{s}}} \right)^T \left(\frac{\partial_R \boldsymbol{\omega}}{\partial \dot{\mathbf{s}}} \right)^T \right]_j \left[\begin{array}{c} {}_R \dot{\mathbf{p}} + {}_R \tilde{\boldsymbol{\omega}}_{IRRR} \mathbf{p} - {}_R \mathbf{f}^e \\ {}_R \dot{\mathbf{L}} + {}_R \tilde{\boldsymbol{\omega}}_{IRRL} \mathbf{L} - {}_R \mathbf{M}^e \end{array} \right]_j \right\} = 0. \quad (14.9)$$

where each subsystem i consists of N_i bodies and has n_i degrees of freedom. Additionally it is useful to describe each subsystem by a different set of describing velocities $\dot{\mathbf{y}}_i \in \mathbb{R}^{6+n_i}$ which are themselves a function of the generalized velocities $\dot{\mathbf{s}}$. Splitting up the partial derivative of (14.9) yields the Projection Equation in subsystem representation

$$\sum_{i=1}^{N_s} \left(\frac{\partial \dot{\mathbf{y}}_i}{\partial \dot{\mathbf{s}}} \right)^T \left\{ \sum_{j=1}^{N_i} \left[\left(\frac{\partial_R \mathbf{v}_c}{\partial \dot{\mathbf{y}}_i} \right)^T \left(\frac{\partial_R \boldsymbol{\omega}}{\partial \dot{\mathbf{y}}_i} \right)^T \right]_j \left[\begin{array}{c} {}_R \dot{\mathbf{p}} + {}_R \tilde{\boldsymbol{\omega}}_{IRRR} \mathbf{p} - {}_R \mathbf{f}^e \\ {}_R \dot{\mathbf{L}} + {}_R \tilde{\boldsymbol{\omega}}_{IRRL} \mathbf{L} - {}_R \mathbf{M}^e \end{array} \right]_j \right\} = 0. \quad (14.10)$$

After summing up over all N_i bodies j of a subsystem i we obtain a representation for each subsystem and (14.10) can be rewritten to

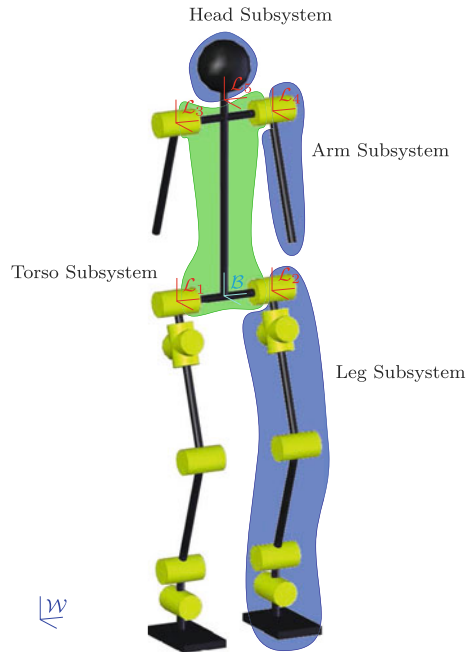
$$\sum_{i=1}^{N_s} \left(\frac{\partial \dot{\mathbf{y}}_i}{\partial \dot{\mathbf{s}}} \right)^T \{ \mathbf{M}_i(\mathbf{q}_i) \ddot{\mathbf{y}}_i + \mathbf{G}_i(\mathbf{q}_i, \dot{\mathbf{y}}_i) \dot{\mathbf{y}}_i - \mathbf{Q}_i(\mathbf{q}_i) \} = 0. \quad (14.11)$$

where $\mathbf{M}_i \in \mathbb{R}^{(6+n_i) \times (6+n_i)}$ represents the mass matrix of the subsystem, $\mathbf{G}_i \in \mathbb{R}^{(6+n_i) \times (6+n_i)}$ summarizes centrifugal and Coriolis forces and $\mathbf{Q}_i \in \mathbb{R}^{6+n_i}$ represents generalized forces and torques acting on the subsystem. The subsystem matrices and vectors are functions of the generalized velocities $\dot{\mathbf{y}}_i$ and the generalized coordinates $\mathbf{q}_i \in \text{SE}(3) \times \mathbb{R}^{n_i}$ of the subsystem. Both of them can be expressed as a function of the generalized velocities $\dot{\mathbf{s}}$ and the generalized coordinates \mathbf{q} of the whole robotic system.

After we found the corresponding matrices \mathbf{M}_i and \mathbf{G}_i and the vector \mathbf{Q}_i from (14.11) by evaluating the terms in the braces in (14.10) for each subsystem, they can be combined to yield the equations of motion of the complete system. The outer sum in (14.10) can be rewritten as

$$\left[\left(\frac{\partial \dot{\mathbf{y}}_1}{\partial \dot{\mathbf{s}}} \right)^T \cdots \left(\frac{\partial \dot{\mathbf{y}}_{N_s}}{\partial \dot{\mathbf{s}}} \right)^T \right] \left[\begin{array}{c} \mathbf{M}_1 \ddot{\mathbf{y}}_1 + \mathbf{G}_1 \dot{\mathbf{y}}_1 - \mathbf{Q}_1 \\ \vdots \\ \mathbf{M}_{N_s} \ddot{\mathbf{y}}_{N_s} + \mathbf{G}_{N_s} \dot{\mathbf{y}}_{N_s} - \mathbf{Q}_{N_s} \end{array} \right] = 0. \quad (14.12)$$

Fig. 14.2 One possible choice of subsystems and frames for a humanoid robot



14.2.2 Assembly Strategy

In the case of the considered biped robot one reasonable choice of a decomposition in subsystems is given by Fig. 14.2. In total four different kinds of subsystems (subsequently referred to as limb subsystems) for the legs, the arms, the head and the torso are defined.

While the leg and the arm subsystem are used twice for the left and the right side of the robot, the other subsystems are used once yielding a total of $N_s = 6$ subsystems for the robot. The free floating body subsystem is the only root of the kinematic topology of the robot, and all other subsystems are attached to this root.

All of the above mentioned subsystems consist of structural elements and actuators, which themselves can be grouped into smaller subsystems. Thus we start with the smallest subsystem consisting of a motor, a gear, and a structural element attached to the driven side of the gear (subsequently referred to as the drive subsystem) as shown in the left part of Fig. 14.3, and compute the subsystem matrices and vectors. Next we adopt the above subsystem with the inertial parameters of the individual drive subsystems of the considered limb and then recursively combine these drive subsystems to the limb subsystem (see Fig. 14.3 in the middle). Then the equations of motion of the complete robot are synthesized by recursively combining the limb subsystems using (14.12). As a result we find the equations of motion of the humanoid robot system with free moving base as given in (14.7).

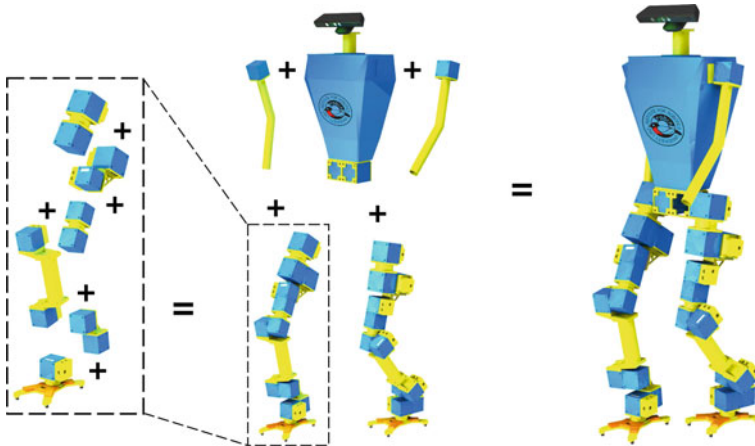
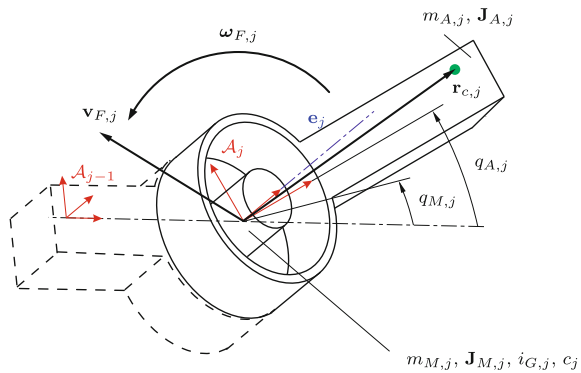


Fig. 14.3 Subsystems are recursively combined to larger subsystems

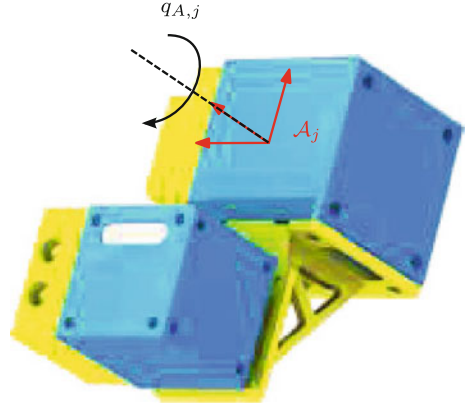
Fig. 14.4 Schematic of a drive subsystem



14.2.3 Drive Subsystem

The drive subsystem is considered the smallest union of structural elements and an actuator. A schematic of the subsystem can be seen in Fig. 14.4 next to a sample representation of the subsystem in Fig. 14.5. It consists of a motor, an elastic gear and an arm. The subsystem has two internal degrees of freedom, the position of the motor and the position of the arm. The arm position is given by $q_{A,j}$, the motor position transferred to the gear output is given by $q_{M,j}$. The gear elasticity is modeled as a rotational spring at the output of the gear. While in (14.10) one can use different arbitrary reference frames R for each body of the subsystem, in most cases it is constructive to use a body fixed frame \mathcal{A}_i attached to the arm as reference frame. In this case the inertia tensor of a rigid body is constant w.r.t. time. The different variables in Fig. 14.4 are defined as follows

Fig. 14.5 Representation of a drive subsystem—Hip roll joint



- $m_{M,j}$: mass of the motor
- $m_{A,j}$: mass of the arm
- $i_{G,j}$: gear ratio
- c_i : linear spring constant of the gear elasticity
- $\mathbf{J}_{M,j}$: inertia tensor of the motor represented in frame \mathcal{A}_j
- $\mathbf{J}_{A,j}$: inertia tensor of the arm represented in frame \mathcal{A}_j
- $\mathbf{r}_{c,j}$: position of the CoM of the arm with respect to \mathcal{A}_j
- \mathbf{e}_j : unit vector aligned with the axis of rotation represented in frame \mathcal{A}_j
- $\mathbf{v}_{F,j}$: translational velocity of the origin of \mathcal{A}_j represented in frame \mathcal{A}_j
- $\boldsymbol{\omega}_{F,j}$: angular velocity of the frame \mathcal{A}_{j-1} represented in frame \mathcal{A}_j
- $q_{M,j}$: angle of rotation of the motor around \mathbf{e}_j divided by the gear ratio i_G
- $q_{A,j}$: angle of rotation of the arm around \mathbf{e}_j .

Using the generalized subsystem velocities²

$$\dot{\mathbf{y}}_{d,j} = [\mathbf{v}_{F,j}^T \ \boldsymbol{\omega}_{F,j}^T \ \dot{\mathbf{q}}_{J,j}^T]^T \quad (14.13)$$

with the vector of motor and arm position

$$\mathbf{q}_{J,j} = [q_{M,j} \ q_{A,j}]^T \in \mathcal{J}_j \quad (14.14)$$

and evaluating the terms in brackets in (14.10) yields the subsystem matrices and vectors for the drive subsystem. In contrast to the equations of motion of the robot, in the subsystem equations we still have to consider reaction forces $\mathbf{Q}_{d,j}^r$ introduced by the neighboring joint units

$$\mathbf{M}_{d,j} \ddot{\mathbf{y}}_{d,j} + \mathbf{G}_{d,j} \dot{\mathbf{y}}_{d,j} - \mathbf{Q}_{d,j}^e = \mathbf{Q}_{d,j}^r. \quad (14.15)$$

²Index d stands for drive subsystem.

14.2.4 Limb Subsystem

Each limb subsystem consists of N_l drive subsystems.³ The generalized velocities of a limb subsystem are chosen as

$$\dot{\mathbf{y}}_{l,i} = \left[\mathbf{V}_{l,i}^b{}^T \dot{\mathbf{q}}_{l,i}^T \right]^T \quad (14.16)$$

where $\mathbf{V}_{l,i}^b$ is the twist of the origin of the body fixed frame \mathcal{L}_i , $i \in 1, 2$ attached to the torso, see Fig. 14.2. The vector $\mathbf{q}_{l,i} \subset \mathcal{J}$ combines all internal degrees of freedom of the particular limb subsystem. The various drive subsystems are combined to

$$\left[\left(\frac{\partial \dot{\mathbf{y}}_{d,1}}{\partial \dot{\mathbf{y}}_{l,i}} \right)^T \cdots \left(\frac{\partial \dot{\mathbf{y}}_{d,N_l}}{\partial \dot{\mathbf{y}}_{l,i}} \right)^T \right] \begin{bmatrix} \mathbf{M}_{d,1} \ddot{\mathbf{y}}_{d,1} + \mathbf{G}_{d,1} \dot{\mathbf{y}}_{d,1} - \mathbf{Q}_{d,1}^e \\ \vdots \\ \mathbf{M}_{d,N_l} \ddot{\mathbf{y}}_{d,N_l} + \mathbf{G}_{d,N_l} \dot{\mathbf{y}}_{d,N_l} - \mathbf{Q}_{d,N_l}^e \end{bmatrix} = \mathbf{Q}_{l,i}^r \quad (14.17)$$

and yield the subsystem equations

$$\mathbf{M}_{l,i} \ddot{\mathbf{y}}_{l,i} + \mathbf{G}_{l,i} \dot{\mathbf{y}}_{l,i} - \mathbf{Q}_{l,i}^e = \mathbf{Q}_{l,i}^r \quad (14.18)$$

with the limb subsystem matrices and vectors. Again reaction forces $\mathbf{Q}_{l,i}^r$ have to be considered on the right hand side of (14.17).

14.2.5 External Forces

External forces, such as motor torques, gravity or contact forces are part of the generalized forces \mathbf{Q} in (14.7). While the motor torques and gravity forces are already included in the subsystems, the ground contact forces are inserted by means of the principal of virtual work afterward. The generalized forces \mathbf{Q} can be separated into one part representing the contact forces \mathbf{Q}_c and another one for the remaining forces \mathbf{Q}_r ,

$$\mathbf{Q} = \mathbf{Q}_c + \mathbf{Q}_r. \quad (14.19)$$

The vector \mathbf{Q}_c is calculated as

$$\mathbf{Q}_c = \sum_j \kappa_j \left(\frac{\partial \mathbf{V}_j^b}{\partial \dot{\mathbf{s}}} \right)^T \mathbf{W}_j \quad (14.20)$$

with \mathbf{V}_j^b as the body-twist of a reference frame \mathcal{F}_j , $j \in L, R$ fixed within the j th contact area and \mathbf{W}_j as the contact wrench acting at the reference frame. The homogenous

³Index l stands for limb subsystem.

transformation of the reference frame w.r.t. the world fixed frame is given by $\mathbf{H}_j = (\mathbf{r}_j, \mathbf{R}_j)$. Contacts can be opened and closed by the scalars κ_j which are equal to one if the contact is active or zero otherwise.

Due to the unilateral contact between the robot and the contact areas, the contact wrenches \mathbf{W}_j have to be limited. Aligning the z -axis of the frame \mathcal{F}_j with the surface normal and placing it in the middle of the corresponding contact area simplifies the definition of these constraints. For a rectangular shaped contact area, like a foot, the constraints can be expressed by

$$0 \leq \mathbf{e}_3^T \mathbf{W}_j \quad (14.21)$$

$$-\frac{b}{2} \mathbf{e}_3^T \mathbf{W}_j \leq \mathbf{e}_4^T \mathbf{W}_j \leq \frac{b}{2} \mathbf{e}_3^T \mathbf{W}_j \quad (14.22)$$

$$-\frac{a}{2} \mathbf{e}_3^T \mathbf{W}_j \leq \mathbf{e}_5^T \mathbf{W}_j \leq \frac{a}{2} \mathbf{e}_3^T \mathbf{W}_j, \quad (14.23)$$

with \mathbf{e}_i as the i th unit vector and a and b as the length and the width of the foot, see Fig. 14.6. As one can see, the perpendicular contact force ($\mathbf{e}_3^T \mathbf{W}_j$) must always be positive, as the various contact areas can only push on the ground. The limitation of the horizontal torques ($\mathbf{e}_4^T \mathbf{W}_j$ and $\mathbf{e}_5^T \mathbf{W}_j$) are a function of the vertical contact forces and the geometry of the foot.

In Fig. 14.1 contact wrenches and the corresponding contact areas are shown exemplary for a biped robot with two contact points $j \in \{L, R\}$ at the feet. The frames are aligned within the contact area as mentioned above.

14.2.6 Equations of Motion

After one has found the subsystem matrices and vectors for the various limb subsystems, they can now be projected to the equations of motion in minimal representation, of a robot with free floating base by applying (14.12) leading to

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{s}} + \mathbf{h}(\dot{\mathbf{s}}, \mathbf{q}) = \begin{pmatrix} 0 \\ \mathbf{B} \end{pmatrix} \boldsymbol{\tau}_J + \sum_j \kappa_j \underbrace{\begin{bmatrix} \mathbf{Ad}_{H_{jB}}^T \\ \mathbf{J}_{Bj}^T \end{bmatrix}}_{\mathbf{J}_j^T} \mathbf{W}_j \quad (14.24)$$

where $\boldsymbol{\tau}_J$ are the joint torques, \mathbf{h} summarizes all nonlinear terms and \mathbf{B} is the input mapping matrix. Contact wrenches are mapped into the configuration space by the Jacobians $\mathbf{J}_j \in \mathbb{R}^{6 \times (n+6)}$. The transpose adjoint matrix for the homogeneous transformation \mathbf{H}_{jB} is given by $\mathbf{Ad}_{H_{jB}}^T$.

Additionally, the unilateral contact between the feet and the ground in the form of the constraints from (14.21) to (14.23) for the contact wrenches has to be considered.

14.2.7 Equations of Motion W.r.t Center of Mass

For the derivation of the equations of motion an intuitive choice of the generalized velocities is given by (14.2). Using the CoM velocity and the angular torso velocity instead of the torso twist gives many benefits for controller design. Switching generalized velocities from (14.2) to (14.5) is realized by a projection of (14.10) to new minimal velocities $\dot{\mathbf{z}}_1$ with the Jacobian $\mathbf{J}_1 = \partial \dot{\mathbf{s}} / \partial \dot{\mathbf{z}}_1$, see [21, 26] for details. The resulting equations of motion with CoM velocity and floating base are

$$\begin{bmatrix} m\mathbf{I} & 0 & 0 \\ 0 & \mathbf{M}_{\omega_B}(\mathbf{q}_J) & \mathbf{M}_{\omega_B,J}(\mathbf{q}_J) \\ 0 & \mathbf{M}_{\omega_B,J}^T(\mathbf{q}_J) & \mathbf{M}_J(\mathbf{q}_J) \end{bmatrix} \dot{\mathbf{z}}_1 + \begin{bmatrix} -m\mathbf{g} \\ \mathbf{h}_{\omega_B}(\dot{\mathbf{z}}_1, \mathbf{q}_J) \\ \mathbf{h}_J(\dot{\mathbf{z}}_1, \mathbf{q}_J) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{B} \end{bmatrix} \tau_J + \sum_j \kappa_j \begin{bmatrix} [\mathbf{R}_j \ 0] \\ \mathbf{J}_{\omega_B,j}^T \\ \mathbf{J}_{J,j}^T \end{bmatrix} \mathbf{W}_j. \quad (14.25)$$

The gravity vector is given by $\mathbf{g} \in \mathbb{R}^3$, m denotes the total weight of the considered system and \mathbf{h}_{ω_B} and \mathbf{h}_J represent the centrifugal and Coriolis forces.

14.3 Reduced Dynamic Models

In the previous section we derived a multi-body model for the robot with elastic gears. In particular for the considered robot this leads to 34 degrees of freedom. This model is adequate for complex and accurate simulations of the robot, especially if one is interested in the dynamics of the drives. For other purposes, like controller design, a reduced model of the robot is necessary. In this section three different reduced models with lower complexity are derived. The first one will assume ideal kinematic gears which reduces the model to 20 degrees of freedom. The second model additionally assumes high gain joint controllers and therefore neglects the joint dynamics leading to 6 degrees of freedom. The last model also disregards the angular momentum and reduces the model to a point mass with 3 degrees of freedom.

While the last two models are used for controller design, the first model can be used for simple simulations to check the stability of the robot.

14.3.1 Ideal Kinematic Gears

For an ideal kinematic gear the velocities of the attached components correspond to the velocity of the motor, therefore the constraint

$$\dot{\Phi} = \begin{pmatrix} \dot{q}_{A,1} - \dot{q}_{M,1} \\ \vdots \\ \dot{q}_{A,n} - \dot{q}_{M,n} \end{pmatrix} = 0 \quad (14.26)$$

and (14.7) follows to

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{s}} + \mathbf{G}(\mathbf{q}, \dot{\mathbf{s}})\dot{\mathbf{s}} - \mathbf{Q}(\mathbf{q}) - \left(\frac{\partial \dot{\Phi}}{\partial \dot{\mathbf{s}}}\right)^T \lambda = 0. \quad (14.27)$$

Using the orthogonality relation (see [2]) results in

$$\left(\frac{\partial \dot{\mathbf{s}}}{\partial \dot{\bar{\mathbf{s}}}}\right)^T [\mathbf{M}(\mathbf{q})\ddot{\mathbf{s}} + \mathbf{G}(\mathbf{q}, \dot{\mathbf{s}})\dot{\mathbf{s}} - \mathbf{Q}(\mathbf{q})] = 0 \quad (14.28)$$

where $\dot{\bar{\mathbf{s}}}$ conveniently is chosen as a subset of $\dot{\mathbf{s}}$ (e.g. by removing the motor velocity).

Finally one obtains the reduced equations of motion in minimal representation to

$$\bar{\mathbf{M}}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{s}}} + \bar{\mathbf{G}}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{s}}})\dot{\bar{\mathbf{s}}} - \bar{\mathbf{Q}}(\bar{\mathbf{q}}) = 0 \quad (14.29)$$

with

$$\bar{\mathbf{q}} \subset \mathbf{q}, \quad (14.30)$$

$$\dot{\bar{\mathbf{s}}} = \frac{\partial \dot{\mathbf{s}}}{\partial \dot{\bar{\mathbf{s}}}} \dot{\bar{\mathbf{s}}} := \mathbf{F}\dot{\bar{\mathbf{s}}}, \quad (14.31)$$

$$\bar{\mathbf{M}} = \mathbf{F}^T \mathbf{M} \mathbf{F}, \quad (14.32)$$

$$\bar{\mathbf{G}} = \mathbf{F}^T \mathbf{G} \mathbf{F} + \mathbf{F}^T \mathbf{M} \dot{\mathbf{F}}, \quad (14.33)$$

$$\bar{\mathbf{Q}} = \mathbf{F}^T \mathbf{Q}. \quad (14.34)$$

14.3.2 High-Gain Joint Controller

The joint angles \mathbf{q}_J can easily be stabilized by a simple high-gain position control law. Doing so and using the singular perturbation theory it is possible to reduce the dynamical model to (for more details see [4])

$$\begin{bmatrix} m\mathbf{I} & 0 \\ 0 & \mathbf{M}_{\omega_B} \end{bmatrix} \begin{pmatrix} \ddot{\mathbf{r}}_c \\ \dot{\omega}_B \end{pmatrix} + \begin{pmatrix} -m\mathbf{g} \\ \mathbf{M}_{\omega_B, J} \ddot{\mathbf{q}}_J^d + \mathbf{h}_{\omega_B} \end{pmatrix} = \sum_j \kappa_j \begin{bmatrix} \mathbf{I} & 0 \\ \tilde{\mathbf{r}}_{cj} & \mathbf{I} \end{bmatrix} \begin{pmatrix} \mathbf{f}_j \\ \boldsymbol{\tau}_j \end{pmatrix}. \quad (14.35)$$

where \mathbf{f}_j and $\boldsymbol{\tau}_j$ are the contact forces and torques with respect to \mathcal{W} corresponding to the contact wrench \mathbf{W}_j . The vector from the CoM to the center of the contact area is given by $\mathbf{r}_{cj} = \mathbf{r}_j - \mathbf{r}_c$. While the dynamical model from (14.35) still considers the full multi-body dynamics one has to notice that the reduced model assumes perfect joint tracking ($\mathbf{q}_J^d - \mathbf{q}_J \rightarrow 0$). The superscript d in (14.35) marks desired values. In our special case, the relation for the time derivative of the inertia matrix

$$\dot{\mathbf{M}} = \mathbf{G} + \mathbf{G}^T \quad (14.36)$$

holds and (14.35) can be rewritten in terms of canonical momenta as

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{L}} \end{pmatrix} + \begin{pmatrix} -m\mathbf{g} \\ 0 \end{pmatrix} = \sum_j \kappa_j \begin{bmatrix} \mathbf{I} & 0 \\ \tilde{\mathbf{r}}_{cj} & \mathbf{I} \end{bmatrix} \begin{pmatrix} \mathbf{f}_j \\ \tau_j \end{pmatrix} \quad (14.37)$$

where \mathbf{L} is the overall angular momentum and \mathbf{p} is the total linear momentum given by

$$\begin{pmatrix} \mathbf{p} \\ \mathbf{L} \end{pmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \mathbf{M}(\mathbf{q}_J) \dot{\mathbf{s}} \quad (14.38)$$

with $\mathbf{M}(\mathbf{q}_J)$ is the inertia matrix from (14.25).

14.3.3 Point Mass Model

Neglecting the angular momentum of the robot leads us to an approximation of the robot that treats the multi-body model as a single point mass. The position of the point mass coincides with the position of the CoM of the full robot. The equations of motion for the point mass model is given by

$$m\ddot{\mathbf{r}}_c - m\mathbf{g} = \sum_j \kappa_j \mathbf{f}_j. \quad (14.39)$$

A further simplification of the point mass model can be done by constraining the height of the CoM to a constant value [14]. In this case only two degrees of freedom, given by the horizontal position of the CoM, remain. This simplified model is referred to as 3D-LIPM (linear inverted pendulum mode).

14.4 Control Applications

The derived multi-body model and the various reduced models can be used in a wide range of different control applications. While many walking controllers use the point mass model given by (14.39) for feedback control [7, 15, 20, 25], only a few controllers consider the angular momentum of the robot [6, 13, 19]. Also for walking pattern generation most robots rely on the point mass model from (14.39), as in this case an online generation of the CoM trajectory can be realized [12, 27]. Considering the angular momentum during walking pattern generation can either be computed offline with the full robot dynamics given by (14.25) [5] or online by estimating the angular momentum [3, 18] through equivalent masses e.g. in the legs, or by the model from (14.35). It is clear that depending on the control application the best dynamical model for the specific application has to be chosen. Two different

control applications, one using the model from (14.35) and another one using the dynamics of the multi-body model from (14.24) will demonstrate how the different models can be used for the control tasks.

14.4.1 Feed-Forward Torques

In the transition from (14.25) to (14.35) we assumed a high gain position control law and sufficiently accurate joint position tracking. For the physical robot, the accuracy of joint tracking can be significantly increased by means of torque feed-forward control. To obtain feed-forward joint torques for a given trajectory in the configuration space $(\mathbf{q}, \dot{\mathbf{s}}, \ddot{\mathbf{s}})$ one needs to calculate the inverse dynamics from (14.24). During single support the contact wrench acting on the swing leg is equal to zero, which leads to a unique solution of the inverse dynamics. In contrast the system is over-actuated during double support leading to an infinite number of solutions for the inverse dynamics. To find a solution during double support one needs to find a criterion, for example the minimum of the contact forces, to obtain a favorable solution. Solving a quadratic optimization problem of the form

$$\min_{\{\mathbf{W}_L, \mathbf{W}_R\}} \alpha \mathbf{W}_L^T \boldsymbol{\Omega} \mathbf{W}_L + (1 - \alpha) \mathbf{W}_R^T \boldsymbol{\Omega} \mathbf{W}_R \quad (14.40)$$

$$\text{s.t. } \kappa_L \mathbf{Ad}_{HLB}^T \mathbf{W}_L + \kappa_R \mathbf{Ad}_{HRB}^T \mathbf{W}_R = \mathbf{W}, \quad (14.41)$$

with $\alpha \in [0, 1]$ as a scalar coefficient weighting the contact forces to allow a smooth transition between single and double support and $\boldsymbol{\Omega}$ as diagonal scaling matrix leading to a dimensionless objective function, gives the optimal force distribution that minimizes the Euclidian norm of the contact forces. The vector \mathbf{W} represents the wrench that has to act at the upper-body \mathcal{B} to realize the desired motion and is given by the first six equations of the left hand side of (14.24)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{s}} + \mathbf{h}(\dot{\mathbf{s}}, \mathbf{q}) = \begin{pmatrix} \mathbf{W} \\ \boldsymbol{\tau}_J \end{pmatrix}. \quad (14.42)$$

As already mentioned there is only a unilateral contact between a foot and the ground, which limits the set of feasible contact wrenches and the constraints from (14.21) to (14.23) need to be considered.

By limiting $\mathbf{e}_1^T \mathbf{W}_k$, $\mathbf{e}_2^T \mathbf{W}_k$ and $\mathbf{e}_6^T \mathbf{W}_k$ with $k \in \{L, R\}$ and \mathbf{e}_i as the i th unit vector, according to a physical friction law, slipping between the ground and the feet can be prevented as further constraints. In this work sufficient friction is assumed which has held for all tested walking trajectories in simulations and in real world so far. According to (14.24) the desired joint torques $\boldsymbol{\tau}_J$ can now be calculated from the distributed ground reaction forces. If one would just use κ_L and κ_R to turn the contacts on and off during single and double support this would lead to non-continuous ground reaction forces and therefore to non-continuous joint torques. This can be omitted by choosing an appropriate distribution coefficient α . A good choice can be made

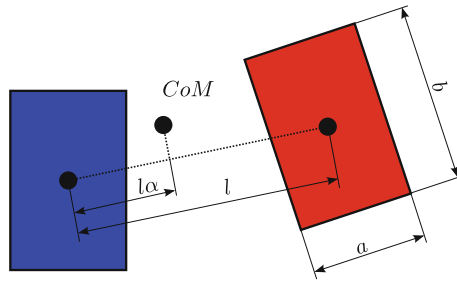


Fig. 14.6 Geometrical interpretation of weighting factor α

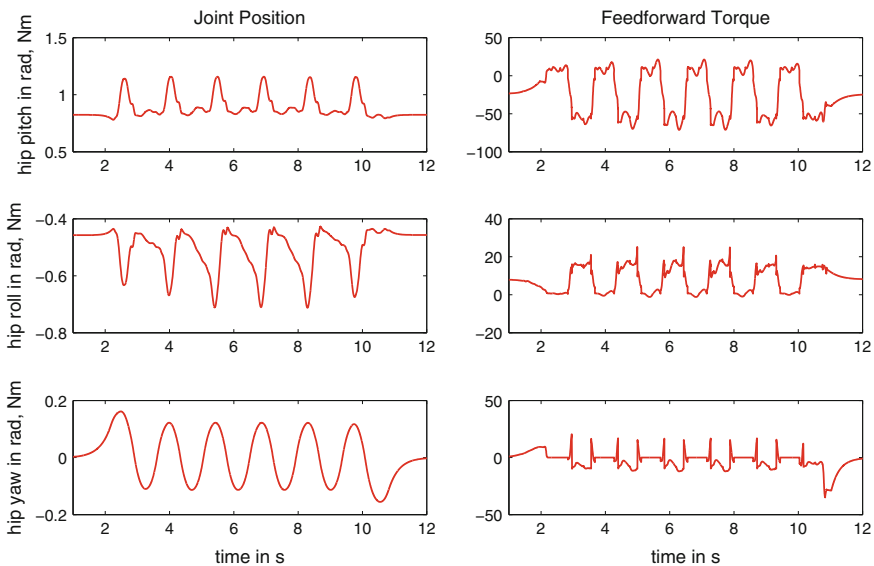


Fig. 14.7 Feed-forward torques for a walking gait exemplarily shown for the *right* hip joints

by the relative position of the CoM relative to the origins of the feet, see Fig. 14.6. Any other rule for the force distribution, like proposed in [15], will lead to similar results.

Figure 14.7 exemplarily shows the feed-forward torques for the hip joints and the corresponding desired joint positions for a typical walking gait where the robot walks forward. In Fig. 14.8 one can see the desired contact forces and torques for the same walking pattern.

14.4.2 Angular Momentum Control

In the walking pattern generation typically reference trajectories for the lower body coordinates are designed according to the dynamic constraints of the unilateral con-

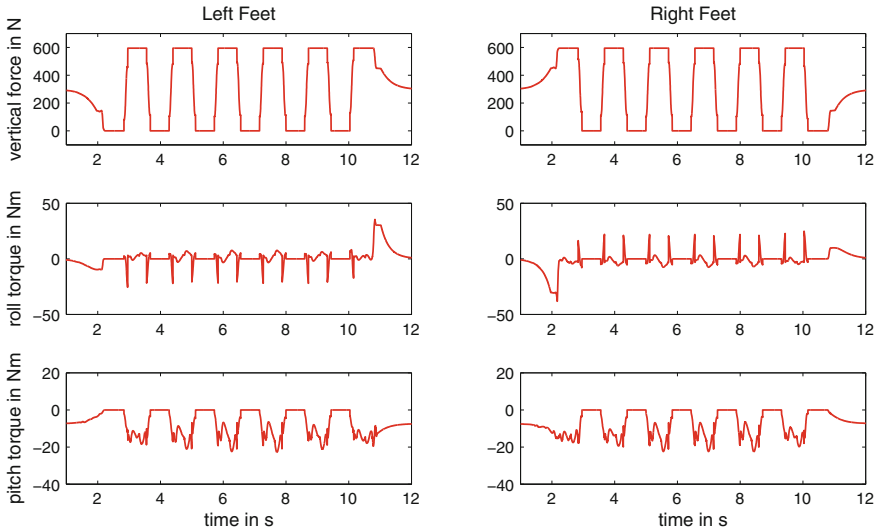


Fig. 14.8 Contact forces and torques for a typical walking gait

tact of the feet to the ground. While in most cases only the overall linear momentum \mathbf{p} from (14.37) of the robot is considered during this trajectory design stage, the neglected angular momentum \mathbf{L} can cause the robot to slip and rotate about its vertical axis [13]. Using the arms, like humans do during walking, to compensate the angular momentum caused by the joints of the lower body can reduce the overall angular momentum significantly.

The total angular and linear momenta of the robot are given by (14.38) where the inertia matrix gives the relation between both momenta and the generalized velocities. While the linear momentum \mathbf{p} is a linear function of the CoM velocity, the overall angular momentum \mathbf{L} depends on the joint positions and the angular velocity.

A short excursion to inverse kinematics calculations shows that the inverse kinematics for a redundant robot can be calculated by using the differential kinematics Jacobian [22, 24]

$$\dot{\mathbf{z}}_i = \mathbf{J}_i \dot{\mathbf{s}} \tag{14.43}$$

with $\dot{\mathbf{z}}_i$ from (14.5) to (14.6) and their corresponding Jacobians. The velocities in the configuration space can be obtained by

$$\dot{\mathbf{s}} = \mathbf{J}_i^{-1} \dot{\mathbf{z}}_i. \tag{14.44}$$

Knowing the initial configuration of the robot, velocities can be summed up over time to get the corresponding configuration of the robot in task space coordinates.

Instead of using the generalized velocities $\dot{\mathbf{z}}_i$ in (14.43) we introduce a new vector

$$\dot{\mathbf{z}}_m = \mathbf{S} \begin{bmatrix} \dot{\mathbf{z}}_i \\ \mathbf{L} \end{bmatrix} = \mathbf{S} \underbrace{\begin{bmatrix} \mathbf{J}_i \\ [0 \ \mathbf{I} \ 0] \mathbf{M} \end{bmatrix}}_{\mathbf{J}_m} \dot{\mathbf{s}} \quad (14.45)$$

which is an extension of the generalized velocities $\dot{\mathbf{z}}_i$ by the total angular momentum. The matrix $\mathbf{S} \in \mathbb{N}^{(6+n) \times (12+n)}$ is a binary selection matrix used to pick the $6 + n$ new coordinates of interest. The total linear momentum cannot be added to (14.45) as the velocity of the CoM and the total linear momentum are linearly dependent. If we want to use the arms (q_{13} and q_{14}) of the robot in Fig. 14.3, with only one degree of freedom per arm, to compensate for the angular momentum around the vertical axis we can define the new vector

$$\dot{\mathbf{z}}_m = (\dot{\mathbf{r}}_c^T \boldsymbol{\omega}_B^T \mathbf{V}_L^{bT} \mathbf{V}_R^{bT} \dot{q}_{13} + \dot{q}_{14} L_z)^T. \quad (14.46)$$

Instead of directly choosing the velocity of the second arm we introduce $\dot{q}_{13} + \dot{q}_{14}$ as relative velocity of the two arms to tie their motion together. This finally gives the selection matrix for the task space velocities \mathbf{z}_2 to

$$\mathbf{S} = \begin{bmatrix} \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{e}_1^T + \mathbf{e}_2^T \\ & & & & 0 \end{bmatrix} & 0 \\ & \mathbf{e}_3^T \end{bmatrix}. \quad (14.47)$$

For the control of biped robots in general we are only interested in the joint angles \mathbf{q}_J for a given trajectory in the configuration space \mathcal{T}_2 . Thus instead of (14.43) we can also use

$$\dot{\mathbf{z}}_m = \mathbf{S} \mathbf{J}_m \mathbf{J}_1^{-1} \dot{\mathbf{z}}_1 \quad (14.48)$$

$$= \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \\ \mathbf{A}_4 & \mathbf{A}_5 & \mathbf{A}_6 \\ 0 & 0 & \mathbf{e}_{13}^T + \mathbf{e}_{14}^T \\ 0 & \mathbf{e}_3^T \mathbf{M}_{\omega_B} & \mathbf{e}_3^T \mathbf{M}_{\omega_B, J} \end{bmatrix} \dot{\mathbf{z}}_1. \quad (14.49)$$

to calculate the inverse kinematics. Adopting the special structure of the Jacobian in (14.49) we get

$$\begin{bmatrix} \mathbf{V}_R^b \\ \mathbf{V}_L^b \\ \dot{q}_{13} + \dot{q}_{14} \\ L_z \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_4 & \mathbf{A}_5 \\ 0 & 0 \\ 0 & \mathbf{e}_3^T \mathbf{M}_{\omega_B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_c \\ \boldsymbol{\omega}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_3 \\ \mathbf{A}_6 \\ \mathbf{e}_{13}^T + \mathbf{e}_{14}^T \\ \mathbf{e}_3^T \mathbf{M}_{\omega_B, J} \end{bmatrix} \dot{\mathbf{q}}_J. \quad (14.50)$$

The joint velocities are then given by

$$\dot{\mathbf{q}}_J = \begin{bmatrix} \mathbf{A}_3 \\ \mathbf{A}_6 \\ \mathbf{e}_{13}^T + \mathbf{e}_{14}^T \\ \mathbf{e}_3^T \mathbf{M}_{\omega_B, J} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{V}_R^b \\ \mathbf{V}_L^b \\ \dot{q}_{13} + \dot{q}_{14} \\ L_z \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_4 & \mathbf{A}_5 \\ 0 & 0 \\ 0 & \mathbf{e}_3^T \mathbf{M}_{\omega_B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_c \\ \boldsymbol{\omega}_B \end{bmatrix} \right). \quad (14.51)$$

With an Euler integration method joint angles can be calculated to

$$\mathbf{q}_J(t_{k+1}) = \mathbf{q}_J(t_k) + \dot{\mathbf{q}}_J(t_k) \Delta t. \quad (14.52)$$

To avoid a numerical drift a stabilization term is added to the general formulation of the inverse kinematics given by (14.51) leading to

$$\dot{\mathbf{q}}_J = \begin{bmatrix} \mathbf{A}_3 \\ \mathbf{A}_6 \\ \mathbf{e}_{13}^T + \mathbf{e}_{14}^T \\ \mathbf{e}_3^T \mathbf{M}_{\omega_B, J} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{V}_R^b \\ \mathbf{V}_L^b \\ \dot{q}_{13} + \dot{q}_{14} \\ L_z \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_4 & \mathbf{A}_5 \\ 0 & 0 \\ 0 & \mathbf{e}_3^T \mathbf{M}_{\omega_B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_c \\ \boldsymbol{\omega}_B \end{bmatrix} + \begin{bmatrix} \mathbf{K}_R & 0 & 0 \\ 0 & \mathbf{K}_L & 0 \\ 0 & 0 & K_A \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}_R \\ \mathbf{e}_L \\ e_A \end{bmatrix} \right) \quad (14.53)$$

with \mathbf{K}_i as positive definite gain matrices the arm position error $e_A = q_{13} + q_{14}$ and the position and orientation errors of the feet

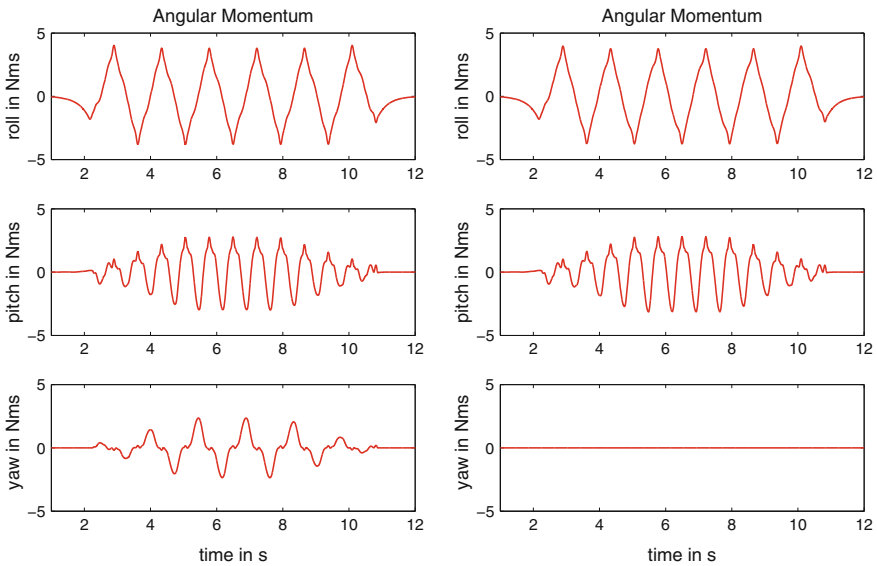


Fig. 14.9 Time evolution of the angular momentum for a typical walking gait: (*left*) without angular momentum compensation, (*right*) with angular momentum compensation

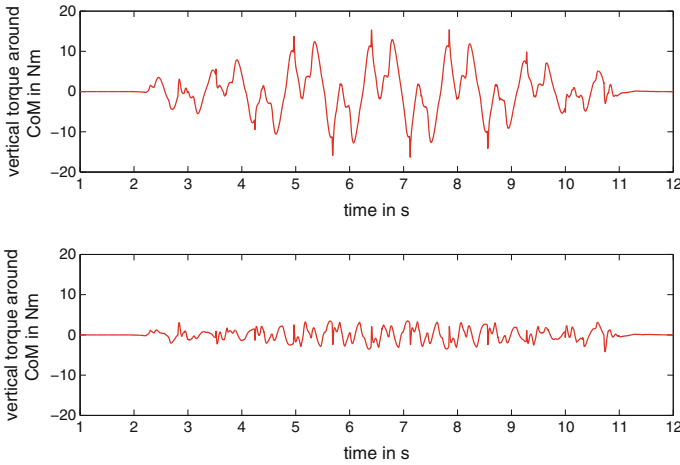
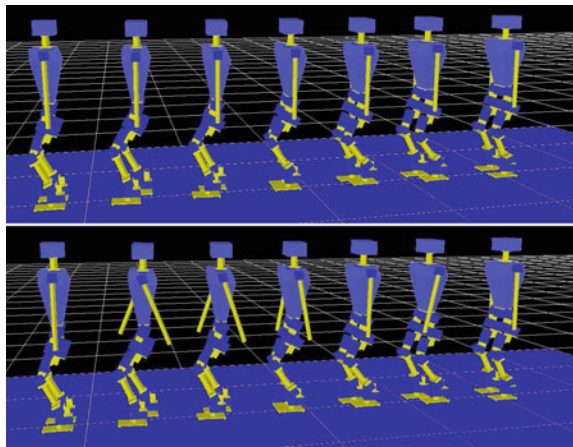


Fig. 14.10 Time evolution of the vertical torque around the CoM measured by the force/torque sensors in the feet: (*top*) without angular momentum compensation, (*bottom*) with angular momentum compensation

Fig. 14.11 Snapshots of the robot walking: (*top*) without angular momentum compensation, (*bottom*) with angular momentum compensation



$$\mathbf{e}_j = \begin{bmatrix} \mathbf{r}_j^d - \mathbf{r}_j \\ \sin(\varphi_j)\mathbf{u}_j \end{bmatrix} \quad j \in \{L, R\} \tag{14.54}$$

with \mathbf{u}_j as the axis of rotation and φ_j as the rotation angle of the matrix $\mathbf{R}_j^e = \mathbf{R}_j^d \mathbf{R}_j^T$. A more detailed analysis of the numerical stabilization of the inverse kinematics with the geometric Jacobian for different orientation representations can be found in [23, 24].

Figure 14.9 shows the time evolution of the angular momentum generated once with a classical inverse kinematics and once generated with the proposed angular momentum control. It can be seen that the vertical angular momentum is constant

at zero if the compensation is turned on. While Fig. 14.9 presents the feed-forward controlled angular momentum, Fig. 14.10 depicts the torque around the CoM for a simulated robot. Obviously, the torque around the CoM is significantly reduced by the proposed angular momentum compensation. Snapshots of the walking robot are shown in Fig. 14.11.

14.5 Conclusions

In this contribution we presented a detailed dynamic model that can be used for the control of humanoid robots. The derivation of the dynamical model uses various subsystems combined recursively to the equations of motion. The used methods allow to easily adopt the kinematic structure of single limbs and to reuse results obtained for limbs with similar kinematic structures but different inertial parameters such as for symmetric legs. After we found a recursive formulation to calculate the equations of motion we perform various state transformations and apply some model simplifications to gain equations that can be used to solve control problems effectively. Next to the modeling of the robot we presented two control applications that use the derived model simplifications. For feed-forward torque control inverse dynamics is used together with a parameter optimization to calculate necessary joint torques. Further on we extended the inverse kinematics algorithm of the robot to resolve a desired angular momentum to a motion of the robot. Simulations showed the effectiveness of the proposed algorithm.

Acknowledgments This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

References

1. Bremer H (2003) On the use of nonholonomic variables in robotics. In: Belayev A, Guran A (eds) Selected topics in structronics and mechatronic systems. Springer, Heidelberg
2. Bremer H (2008) Elastic multibody dynamics: a direct ritz approach. Springer, Heidelberg
3. Buschmann T, Lohmeier S, Bachmayer M, Ulbrich H, Pfeiffer F (2007) A collocation method for real-time walking pattern generation. In: Proceedings of the IEEE/RAS international conference on humanoid robots, December 2007. pp 1–6
4. Buschmann T, Lohmeier S, Ulbrich H (2009) Biped walking control based on hybrid position/force control. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, October 2009. pp 3019–3024
5. Buschmann T, Lohmeier S, Ulbrich H, Pfeiffer F (2005) Optimization based gait pattern generation for a biped robot. In: Proceedings of the IEEE/RAS international conference on humanoid robots, December 2005. pp 98–103
6. Engelsberger J, Ott C (2012) Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking. In: Proceedings of the IEEE-RAS international conference on humanoid robots, November 2012. pp 183–189

7. Engelsberger J, Ott C, Roa MA, Albu-Schäffer A, Hirzinger G (2011) Bipedal walking control based on capture point dynamics. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, September 2011. pp 4420–4427
8. Featherstone R (1983) The calculation of robot dynamics using articulated-body inertias. *Int J Robot Res* 2(1):13–30
9. Gatringer H, Bremer H, Kastner M (2011) Efficient dynamic modeling for rigid multi-body systems with contact and impact. *Acta Mech* 219:111–128
10. Hollerbach John M (1980) A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans Syst, Man, Cybern* 10:730–736
11. Hyon S, Hale JG, Cheng G (2007) Full-body compliant human humanoid interaction: balancing in the presence of unknown external forces. *IEEE Trans Robot* 23(5):884–898
12. Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K, Hirukawa H (2003) Biped walking pattern generation by using preview control of zero-moment point. In: Proceedings of the international conference on robotics and automation. pp 1620–1626
13. Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K, Hirukawa H (2003) Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, October 2003. pp 1644–1650
14. Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H (2001) The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. pp 239–246
15. Kajita S, Morisawa M, Miura K, Nakaoka S, Harada K, Kenji K, Kanehiro F, Yokoi K (2010) Biped walking stabilization based on linear inverted pendulum tracking. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, October 2010. pp 1189–4496
16. Khalil W (2010) Dynamic modeling robots using recursive newton-euler techniques. In: Cetto J, Filipe J, Ferrier J (eds) Proceedings of the 7th international conference on informatics in control, automation and robotics, vol 2. SciTePress, INSTICC, IFAC, pp 19–31
17. Luh JYS, Walker MW, Paul RPC (1980) On-line computational scheme for mechanical manipulators. *ASME J Dyn, Syst, Meas Control* 102:69–76
18. Mayr J, Gatringer H, Bremer H (2012) A bipedal walking pattern generator that considers multi-body dynamics by angular momentum estimation. In: Proceedings of the IEEE/RAS international conference on humanoid robots, December 2012. pp 178–182
19. Mayr J, Gatringer H, Bremer H (2013) Bipedal balancing control based on the centroidal momentum pivot and the best com-cmp regulator. In: Proceedings of 39th annual conference of the IEEE industrial electronics society, November 2013. pp 4059–4064
20. Nishiwaki K, Kagami S (2006) High frequency walking pattern generation based on preview control of zmp. In: Proceedings of the IEEE international conference on robotics and automation, May 2006. pp 2667–2672
21. Ott C, Roa MA, Hirzinger G (2011) Posture and balance control for biped robots based on contact force optimization. In: Proceedings of the IEEE/RAS international conference on humanoid robots, October 2011. pp 26–33
22. Sciavicco L, Siciliano B (2004) Modelling and control of robot manipulators. Springer, United Kingdom
23. Siciliano B, Villani L (1999) Robot force control. Robotics: vision, manipulation and sensors. Kluwer Academic Publishers, Boston
24. Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics—Modelling, Planning and Control. Advanced textbooks in control and signal processing series. Springer, Heidelberg
25. Takenaka T, Matsumoto T, Yoshiike T, Hasegawa T, Shirokura S, Kaneko H, Orita A (2009) Real time motion generation and control for biped robot -4th report: Integrated balance control. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, October 2009. pp 1601–1608

26. Wieber P-B (2006) Holonomy and nonholonomy in the dynamics of articulated motion. Fast motions in biomechanics and robotics, Lecture notes in control and information sciences, Springer, Berlin
27. Wieber P-B (2006) Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In: Proceedings of the IEEE/RAS international conference on humanoid robots