Hubert Gattringer
Johannes Gerstmayr *Editors*

# Multibody System Dynamics, Robotics and Control

Springer

Multibody System Dynamics, Robotics and Control

Hubert Gattringer • Johannes Gerstmayr
Editors

# Multibody System Dynamics, Robotics and Control

Springer

*Editors*
Hubert Gattringer
Institute for Robotics
Johannes Kepler University Linz
Linz
Austria

Johannes Gerstmayr
Linz Center of Mechatronics GmbH
Linz
Austria

ISBN 978-3-7091-1288-5 ISBN 978-3-7091-1289-2 (eBook)
DOI 10.1007/978-3-7091-1289-2
Springer Wien Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012954007



Printed on acid-free paper

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

# Preface

This book contains the contributions presented during the *Workshop on Multibody System Dynamics, Robotics and Control*, which took place at the Johannes Kepler University of Linz, Austria, in September 2011. The workshop aimed at bringing together international scientists with an outstanding expertise in mechanics and control, with emphasis on the application to advanced machines and robotic systems. The international character of the workshop was deepened by the participation of widely renowned scientists from Europe. The workshop continued a series of international workshops, which started with the *Japan-Austria Joint Workshop on Mechanics and Model Based Control of Smart Materials and Structures* in September 2008 and the *Russia-Austria Joint Workshop on Advanced Dynamics and Model Based Control of Structures and Machines* in April 2010; both took place in Linz, Austria.

This series of workshops is organized within the framework of the Area *Mechanics and Model Based Control* of the *Austrian Center of Competence in Mechatronics (ACCM)*. This peer-reviewed center of competence served as the steering organisation for the workshop series. Mechanics and Model Based Control are rapidly expanding scientific fields and fundamental disciplines of engineering, particularly in Mechatronics. They share demanding mathematical and/or system-theoretic formulations and methods. One challenge in Mechanics and Model Based Control is to use the ever-increasing computer power with respect to both the simulation of complex physical phenomena in mechanics and the design and real-time implementation of novel control systems. From a strategic point of view, the key objectives of the workshop series are:

- Enabling the interchange of ideas from multibody system dynamics, robotics and control
- Clarification of expectations of researchers in the field of mechanics from advanced control theory and vice versa
- Development of joint international research proposals and teams
- Encouragement of collaborations among industry and universities across the borders of the participating countries

The main topics of the present *Workshop on Multibody System Dynamics, Robotics and Control* were:

- Time/energy optimal path planning for robotic systems
- Optimization in multibody dynamics
- Novel control concepts for flexible multibody systems and robots
- Humanoid robots
- Mobile robots
- Wire robots
- Vibration control for flexible robots
- Control in biomechanics

We believe that the workshop will finally result into the creation of research teams within europe. Such teams should push the frontiers of advanced dynamics and model-based control of machines and robotic systems to new dimensions, resulting in the advanced design of future applications.

The undersigned editors of the present book entitled *Multibody System Dynamics, Robotics and Control* are happy to present the following 17 full-length papers. It is hoped that these contributions will further stimulate the international research and cooperation in the field. The present book is aimed as a third volume of a future *Series in Research on Advanced Methods of Mechatronics*.

March 2012

*Hubert Gattringer*
*Johannes Gerstmayr*

# Acknowledgements

Support of the *Workshop on Multibody System Dynamics, Robotics and Control* in the framework of the *COMET K2 Austrian Center of Competence in Mechatronics (ACCM)* is gratefully acknowledged. The ACCM is a Research and Innovation Centre for collaborative research between science and industry in the field of mechatronics at an international level with the goal of achieving scientific knowledge and results. ACCM executes an ambitious research program in collaboration with researchers, scientific partners and companies worldwide, and serves as steering organisation of the workshop. ACCM is jointly run by the following organisations:

- *Johannes Kepler University of Linz*
- *Linz Center of Mechatronics GmbH*
- *voestalpine AG*

The help of these organisations is highly appreciated. In particular, the editors wish to thank the Johannes Kepler University for serving as host of the workshop. The workshop would not have been possible without the sponsorship of *City of Linz* and *Salvagnini Maschinenbau GmbH*.

# Contents

# Chapter 1
# Time-Optimal Path Planning Along Specified Trajectories

**Francisco Geu Flores and Andrés Kecskeméthy**

**Abstract** Time-optimal motion planning along specified paths is a well-understood problem in robotics, for which well-established methods exist for some standard effects, such as actuator force limits, maximal path velocity, or sliding friction. This paper describes some extensions of the classical methods which consider, on the one hand side, additional non linear constraints such as sticking friction, acceleration limits at the end-effector, as well power limits for the overall system, and on the other, general paths featuring smooth interpolation of angular acceleration as well as arbitrary multibody systems comprising multiple loops. The methods are illustrated with two applications from robotics and the mining industry.

## 1.1 Introduction

The problem of computing the time-optimal motion of a manipulator along a prescribed spatial path under forces, velocities and acceleration constraints has been thoroughly studied in the past. The basic idea of most solution algorithms is based on the pioneer work by Dubowsky, Bobrow and Gibson [1] as well as Shin and McKay [7], and the modifications proposed by Pfeiffer and Johanni [5] and Shiller and Lu [6]. It consists in mapping both the multibody differential equations and the system constraints to a one-dimensional motion along the prescribed path in order to define the maximally allowed accelerations at each point on the path, and to seek, by means of forwards and backwards integrations, the optimal set of points where the optimal acceleration must switch from a maximum to a minimum or *vice versa*.

F. Geu Flores (✉) • A. Kecskeméthy
Universität Duisburg-Essen, Lotharstr. 1, Duisburg 47057, Germany
e-mail: francisco.geu@uni-due.de; andres.kecskemethy@uni-due.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_1, 

Messner, Gattringer and Bremer [4] extend these ideas to handle jerk constraints. Moreover, by applying methods of optimal control, they avoid searching for the switching points with backwards and forwards integrations, thus rendering very efficient, online-suitable code. However, their method is restricted to non-singular arcs, as well as systems for which the state of vanishing velocity is always feasible, thus limiting both search space and application domain. Although such cases are very special, they do happen in practice, two corresponding examples being shown in this paper.

A further recent approach is the reformulation of the problem as a convex optimal control problem by Verscheure, Demeulenaere, Swevers, De Schutter and Diehl [9]. This approach yields a very efficient and stable offline procedure. However, by considering only constraints that are linear in $\ddot{s}$ and $\dot{s}^2$, the application domain is additionally limited with respect to [7].

In summary, research on optimal-time path planning can be viewed as being focused on two directions. One is to develop new time-optimal algorithms as a basis for finding time-optimal paths between two configurations. In this setting, the aforementioned motion planning algorithm is used as an internal loop for computing the minimal-time motion after each path geometry variation. The second direction is to extend the systems for which time-optimal motion planning can be performed, both from the viewpoint of the multibody modeling as well as from the viewpoint of the regarded constraints.

This paper considers the second aim, extending the robotic modeling environment to arbitrary multiloop systems with general rigid-body motion interpolation options, as well as by adding sticking friction and global power limits to the set of allowable constraints. This allows for applying the method to a broader set of industrial problems and for easily generating the constraints of the path-planning algorithm regardless of the complexity of the kinematical structure. For motion interpolation along a prescribed path, we consider both interpolation along the spatial path between key poses of the end-effector (including smooth angular acceleration interpolation) as well as interpolation in generalized-coordinates (i.e. joint space), between which the user can switch. The flexibility of the method is shown by means of two examples in the mining industry and the field of robotics respectively. Further papers will consider the optimization of path geometry.

## 1.2 Spatial Paths in Multibody Systems

### *1.2.1 Kinetostatic Transmission Elements*

A mechanical system can be regarded as a sequence of kinetostatic transmission elements mapping motion and forces from one set of state objects — the ‘input’ variables $\underline{q}_{\text{in}}$ — to another set of state objects — the ‘output’ variables $\underline{q}_{\text{out}}$ ([3]).

These state objects can be spatial reference frames and/or scalar variables, including associated velocities, accelerations and generalized forces.

The overall transmission behavior of a kinetostatic transmission element comprises a motion transmission traversal consisting of the three sub-operations

$$\begin{aligned} &\underline{\text{position}}: \quad \underline{q}_{\text{out}} = \varphi(\underline{q}_{\text{in}}) \\ &\underline{\text{velocity}}: \quad \dot{\underline{q}}_{\text{out}} = \mathbf{J}_\varphi \, \dot{\underline{q}}_{\text{in}} \\ &\underline{\text{acceleration}}: \ \ddot{\underline{q}}_{\text{out}} = \mathbf{J}_\varphi \, \ddot{\underline{q}}_{\text{in}} + \dot{\mathbf{J}}_\varphi \, \dot{\underline{q}}_{\text{in}} \end{aligned} \tag{1.1}$$

where $\mathbf{J}_\varphi = \partial\varphi/\partial\underline{q}_{\text{in}}$ represents the Jacobian of the element, as well as force transmission traversal consisting, for ideal transmission elements, of the sub-operation:

$$\underline{\text{force}}: \quad \underline{Q}_{\text{in}} = \mathbf{J}_\varphi{}^{\text{T}} \underline{Q}_{\text{out}}. \tag{1.2}$$

Any (passive) physical or mathematical object that maps a set of input state objects to a set of output state objects without loss or increase of mechanical power can be regarded as a kinetostatic transmission element.

### *1.2.2 Spatial Motion Parametrization*

Let a general spatial path be given by the pose of an output frame $\mathcal{K}_{\text{E}} = \mathcal{P}(s) \in \text{SE}(3)$, with the translation part parametrized by vector $\Delta\underline{r}(s)$ and the rotation part described by a rotation matrix $\Delta\mathbf{R}(s)$, both measured with respect to a basis frame $\mathcal{K}_1$. Let the coordinate $s$ be the path length of $\Delta\underline{r}(s)$. The spatial path $\mathcal{P}(s)$ can be modelled as a kinetostatic transmission element mapping the velocity $\dot{s}$ along the path to the end-effector twist $\underline{\text{t}}_{\text{E}}$ and, by duality, the wrench $\underline{\text{w}}_{\text{E}}$ acting at the end-effector to a generalized force $Q_s$ along the path, as shown in Fig. 1.1.

The pose of $\mathcal{K}_{\text{E}}$ can be computed as a function of the pose of the basis frame $\mathcal{K}_1$ and the path coordinate $s$ as

$$\begin{aligned} \mathbf{R}_{\text{E}} &= \mathbf{R}_1 \, \Delta\mathbf{R} \\ \underline{r}_{\text{E}} &= \Delta\mathbf{R}^{\text{T}}(\underline{r}_1 + \Delta\underline{r}), \end{aligned} \tag{1.3}$$

where general vectors are assumed to be decomposed in the target frame and $\mathbf{R}_i$ denotes the rotation matrix transforming coordinates with respect to frame $\mathcal{K}_i$ to coordinates with respect to frame $\mathcal{K}_0$.

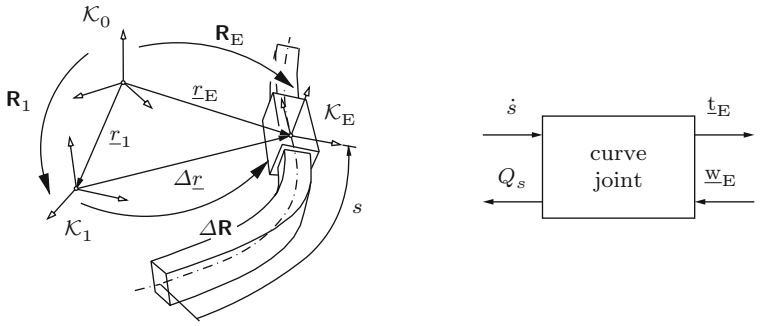



**Fig. 1.1** Spatial joint as a kinetostatic transmission element

The velocity transmission then takes the form

$$\begin{bmatrix} \underline{\omega}_\mathrm{E} \\ \underline{v}_\mathrm{E} \end{bmatrix} = \mathbf{J}_g \begin{bmatrix} \underline{\omega}_1 \\ \underline{v}_1 \end{bmatrix} + \mathbf{J}_\mathcal{P}\,\dot{s}, \qquad \text{with} \quad \mathbf{J}_g = \begin{bmatrix} \Delta\mathbf{R}^\mathrm{T} & \underline{0} \\ -\Delta\mathbf{R}^\mathrm{T}\widetilde{\Delta\underline{r}} & \Delta\mathbf{R}^\mathrm{T} \end{bmatrix}, \tag{1.4}$$

where $\mathbf{J}_g$ is the rigid-body Jacobian, $\mathbf{J}_\mathcal{P}$ is the Jacobian mapping the path velocity $\dot{s}$ along the spatial path to the twist at the output frame $\mathcal{K}_\mathrm{E}$, and $\tilde{\underline{a}}$ refers to the skew-symmetric matrix generated by a three-dimensional vector $\underline{a} = \begin{bmatrix} a_x\, a_y\, a_z \end{bmatrix}^\mathrm{T}$.

The acceleration transmission can be written as

$$\begin{bmatrix} \dot{\underline{\omega}}_\mathrm{E} \\ \underline{a}_\mathrm{E} \end{bmatrix} = \mathbf{J}_g \begin{bmatrix} \dot{\underline{\omega}}_1 \\ \underline{a}_1 \end{bmatrix} + \begin{bmatrix} \underline{0} \\ {}^2\tilde{\underline{\omega}}_1^2 \Delta\mathbf{R}^\mathrm{T}\Delta\underline{r} \end{bmatrix} + \mathbf{J}_\mathcal{P}\,\ddot{s} + \mathbf{J}_\mathcal{P}{}'\dot{s}^2 + \begin{bmatrix} {}^2\widetilde{\underline{\omega}}_1 & \mathbf{0} \\ \mathbf{0} & 2^2\widetilde{\underline{\omega}}_1 \end{bmatrix} \mathbf{J}_\mathcal{P}\,\dot{s}, \tag{1.5}$$

where $(.)'$ denotes a derivative with respect to the path coordinate $s$.

According to Eq. 1.2, the force transmission yields

$$\begin{bmatrix} \underline{\tau}_1 \\ \underline{f}_1 \\ Q_s \end{bmatrix} = \begin{bmatrix} \mathbf{J}_g^\mathrm{T} \\ \mathbf{J}_\mathcal{P}{}^\mathrm{T} \end{bmatrix} \begin{bmatrix} \underline{\tau}_\mathrm{E} \\ \underline{f}_\mathrm{E} \end{bmatrix}. \tag{1.6}$$

A complete description on how $\Delta\underline{r}(s)$ and $\Delta\mathbf{R}(s)$ can be defined can be found in [8]. In the examples presented in this paper, the function $\Delta\underline{r}(s)$ is obtained by interpolating key poses with quintic B-splines, using the DIERCKX curve-fitting routines [2] with prescribed boundary conditions for positions, tangents and curvature. The orientation $\Delta\mathbf{R}(s)$ of the output frame along the curve is then prescribed as a function of the geometry of the curve $\Delta\underline{r}(s)$ and additional elementary rotations with respect to the natural directions of the curve, prescribed as cubic B-spline functions of

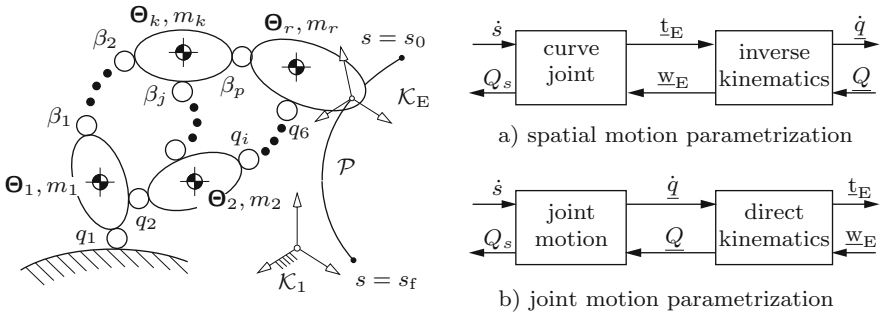



**Fig. 1.2** Motion parametrization using (**a**) target coord. or (**b**) joint coord

the path coordinate $s$. Nevertheless, these equations also hold for more general path parametrizations.

Let $\underline{\varphi}_q$ be the direct kinematics of the manipulator with $n$ degrees of freedom with serial or closed kinematical topology, described by $n$ independent joint coordinates $q_i$, collected in the vector $\underline{q} \in \mathbb{R}^n$. If the spatial path $\mathcal{P}(s)$ of the end-effector is contained in its workspace, the system can be understood as a closed chain with one degree of freedom, as shown in Fig. 1.2a. The corresponding joint motion is, hence, described by the equations

$$\begin{aligned} \underline{q} &= \underline{\varphi}_q{}^{-1}(\mathcal{P}(s)) \\ \dot{\underline{q}} &= \mathbf{J}_\varphi{}^{-1}\mathbf{J}_\mathcal{P}\dot{s} \\ \ddot{\underline{q}} &= \mathbf{J}_\varphi{}^{-1}\mathbf{J}_\mathcal{P}\ddot{s} + \left[\mathbf{J}_\varphi'{}^{-1}\mathbf{J}_\mathcal{P} + \mathbf{J}_\varphi{}^{-1}\mathbf{J}_\mathcal{P}'\right]\dot{s}^2, \end{aligned} \tag{1.7}$$

where $\mathbf{J}_\varphi = \partial \underline{\varphi}_q / \partial \underline{q}$ represents the transmission Jacobian of the manipulator.

### *1.2.3 Joint Motion Parametrization*

For some applications where the interpolated pose of the end-effector does not have to fulfill any geometrical constraints along the given spatial path, or for the special case of manipulators with limited degrees of freedom, the interpolation of motion in joint coordinates can be of advantage. In these cases, the motion can be given as $\underline{q} = \underline{\eta}(s) \in \mathbb{R}^n$, which can also be represented as a kinetostatic transmission element mapping the motion progress, described by a motion coordinate $s(t)$ which is now not the path coordinate, to the joint motion $\underline{q}(t)$ and end-effector motion $\mathcal{K}_E(t)$, as shown in Fig. 1.2b.

In this case the corresponding end-effector motion is described by the equations

$$\begin{aligned} \mathcal{K}_{\mathrm{E}} &= \underline{\varphi}_q\big(\underline{\eta}(s)\big) \\ \mathrm{t}_{\mathrm{E}} &= \mathbf{J}_\varphi \mathbf{J}_\eta \dot{s} \\ \dot{\mathrm{t}}_{\mathrm{E}} &= \mathbf{J}_\varphi \mathbf{J}_\eta \ddot{s} + \left[\mathbf{J}'_\varphi \mathbf{J}_\eta + \mathbf{J}'_\varphi \mathbf{J}_\eta\right]\dot{s}^2, \end{aligned} \tag{1.8}$$

where $\mathbf{J}_\eta = \partial\underline{\eta}/\partial s$ represents the transmission Jacobian of $\underline{\eta}$ with respect to the motion coordinate $s$.

For both cases described above, the velocities and accelerations at the independent manipulator joints and the end-effector have the general form

$$\begin{aligned} \dot{\underline{q}} &= {}_q\mathbf{J}_s\,\dot{s}: & \underline{\mathrm{t}}_{\mathrm{E}} &= {}_{\mathrm{E}}\mathbf{J}_s\,\dot{s}, \\ \ddot{\underline{q}} &= {}_q\mathbf{J}_s\,\ddot{s} + {}_q\mathbf{J}'_s\,\dot{s}: & \dot{\underline{\mathrm{t}}}_{\mathrm{E}} &= {}_{\mathrm{E}}\mathbf{J}_s\,\ddot{s} + {}_{\mathrm{E}}\mathbf{J}'_s\,\dot{s}. \end{aligned} \tag{1.9}$$

## 1.3 Time-Optimal Motion Generation

### *1.3.1 Formulation of the Time-Optimal Problem*

Let the dynamics of a multibody system be described by the differential equations in minimal form

$$\mathbf{M}(\underline{q})\ddot{\underline{q}} + \underline{b}(\underline{q}, \dot{\underline{q}}) + \underline{Q}_{\mathrm{e}}(\underline{q}, \dot{\underline{q}}) + \underline{Q}_{\mathrm{G}}(\underline{q}) = -\underline{Q}, \tag{1.10}$$

where $\mathbf{M}$ is the $n \times n$ mass matrix of the multibody system, $\underline{b}(\underline{q}, \dot{\underline{q}})$ is the $n$-dimensional vector containing the centripetal and Coriolis terms, $\underline{Q}_{\mathrm{G}}(\underline{q})$ is an $n$-dimensional vector containing the projection of the gravitational forces on the generalized coordinates, $\underline{Q}_{\mathrm{e}}(\underline{q}, \dot{\underline{q}})$ is an $n$-dimensional vector containing the projection of general external forces, and $\underline{Q}$ is an $n$-dimensional vector collecting the generalized actuator forces.

Let the velocities $\dot{\underline{q}}$, accelerations $\ddot{\underline{q}}$, and generalized actuator forces $\underline{Q}$ at the joints be constrained by equations of the form

$$\begin{aligned} \dot{\underline{q}}^{\min}\left(\underline{q}\right) &\le \dot{\underline{q}} \le \dot{\underline{q}}^{\max}(\underline{q}) \\ \ddot{\underline{q}}^{\min}(\underline{q}, \dot{\underline{q}}) &\le \ddot{\underline{q}} \le \ddot{\underline{q}}^{\max}(\underline{q}, \dot{\underline{q}}) \\ \underline{Q}^{\min}(\underline{q}, \dot{\underline{q}}) &\le \underline{Q} \le \underline{Q}^{\max}(\underline{q}, \dot{\underline{q}}), \end{aligned} \tag{1.11}$$

and the velocities $\underline{\mathrm{t}}_{\mathrm{E}}$ and accelerations $\dot{\underline{\mathrm{t}}}_{\mathrm{E}}$ at the end-effector be constrained by equations of the form

$$\begin{aligned} \underline{\mathrm{t}}_{\mathrm{E}}^{\min}(\mathcal{K}_{\mathrm{E}}) \leq \underline{\mathrm{t}}_{\mathrm{E}} \leq \underline{\mathrm{t}}_{\mathrm{E}}^{\max}(\mathcal{K}_{\mathrm{E}}) \\ \dot{\underline{\mathrm{t}}}_{\mathrm{E}}^{\min}(\mathcal{K}_{\mathrm{E}},\, \underline{\mathrm{t}}_{\mathrm{E}}) \leq \dot{\underline{\mathrm{t}}}_{\mathrm{E}} \leq \dot{\underline{\mathrm{t}}}_{\mathrm{E}}^{\max}(\mathcal{K}_{\mathrm{E}},\, \underline{\mathrm{t}}_{\mathrm{E}}). \end{aligned} \tag{1.12}$$

With the relations Eq. 1.9, the equations of motion described in Eq. 1.10 can be written in terms of the motion coordinate $s$ as

$$\underline{m}(s)\ddot{s} + \underline{c}(s,\, \dot{s}) + \underline{d}(s) = -\underline{Q}, \tag{1.13}$$

with

$$\begin{aligned} \underline{m}(s) &= \mathbf{M}(s)\, {}_{q}\mathbf{J}_{s} \\ \underline{c}(s,\, \dot{s}) &= \left[\mathbf{M}(s)\, {}_{q}\mathbf{J}_{s}^{'} + \bar{\underline{b}}(s)\right]\, \dot{s}^2 + \underline{Q}_{\mathrm{e}}(s, \dot{s}) \\ \underline{d}(s) &= \underline{Q}_{\mathrm{G}}(s), \end{aligned} \tag{1.14}$$

where the coefficients $m_i$ and $c_i$ represent the effective inertia and velocity forces at every independent joint respectively, and the term $\underline{b}(\underline{q}, \dot{\underline{q}})$ in Eq. 1.10 can be written as $\bar{\underline{b}}(s)\ \dot{s}^2$, with $\bar{\underline{b}}(s)$ depending only on the configuration $s$.

Furthermore, Eq. 1.9 allows for all constraints of the form described in Eqs. 1.11 and 1.12 to be collected in the vector inequality

$$\hat{\underline{b}}_1(s, \dot{s}) \leq \hat{\underline{m}}(s)\, \ddot{s} \leq \hat{\underline{b}}_2(s, \dot{s}), \tag{1.15}$$

where $\hat{\underline{b}}_1$, $\hat{\underline{b}}_2$ and $\hat{\underline{m}}$ are vectors in $\mathbb{R}^l$ and $l$ is the number of constraints.

The left and right terms of this inequality define the set of admissible states $[s,\, \dot{s}]^{\mathrm{T}}$ and can be written as the scalar inequality

$$g(s, \dot{s}) \leq 0, \tag{1.16}$$

with $g(s, \dot{s}) = \max\left\{\hat{b}_{1j}\ (s, \dot{s}) - \hat{b}_{2j}\ (s, \dot{s})\right\}$, for all $j = 1, 2, \cdots, \ell$.

For all constraints $j$ for which $\hat{m}_j(s)$ does not vanish, Eq. 1.15 further limits the acceleration $\ddot{s}$ along the spatial path, since it must hold

$$\frac{\hat{b}_{1j}(s, \dot{s})}{|\hat{m}_j(s)|} \leq \operatorname{sgn}\left[\hat{m}_j(s)\right]\ddot{s} \leq \frac{\hat{b}_{2j}(s, \dot{s})}{|\hat{m}_j(s)|}, \tag{1.17}$$

or compactly

$$l_j(s, \dot{s}) \leq \ddot{s} \leq u_j(s, \dot{s}), \tag{1.18}$$

where $l_j(s,\dot{s})$ and $u_j(s,\dot{s})$ are the lower and upper bounds of the $j$-th constraint, functions of the state $[s,\dot{s}]^{\mathrm{T}}$. These equations can be rewritten as the one dimensional inequality

$$L(s,\dot{s}) \leq \ddot{s} \leq U(s,\dot{s}), \tag{1.19}$$

where $L(s,\dot{s}) = \max\{l_j(s,\dot{s})\}$ and $U(s,\dot{s}) = \min\{u_j(s,\dot{s})\}$ for all $j = 1,2,\cdots,\ell$ for which $\hat{m}_j(s) \neq 0$.

The time optimal problem consists in finding the monotonically increasing function $s(t)$ which minimizes the total time needed to travel from $s_0$ to $s_\mathrm{f}$, without violating Eq. 1.19 or Eq. 1.16.

In this paper, further constraints that match Eq. 1.15 but are nonquadratically nonlinear in $\dot{s}$ are investigated. These stem from linear constraints at the end-effector accelerations in the form

$$|\mathbf{A}(\mathcal{K}_\mathrm{E})\,\dot{\underline{\mathrm{t}}}_\mathrm{E}(\mathcal{K}_\mathrm{E},\underline{\mathrm{t}}_\mathrm{E})| \leq \dot{\underline{\mathrm{t}}}_\mathrm{E}^{\max}, \tag{1.20}$$

where $\mathbf{A}(\mathcal{K}_\mathrm{E})$ is a $6\times 6$ matrix depending on the manipulator configuration, as well as a limit for maximally allowed power consumption

$$|\underline{\dot{q}}^{\mathrm{T}}(\underline{q},\underline{\dot{q}})\underline{Q}(\underline{q},\underline{\dot{q}},\underline{\ddot{q}})| \leq P_{\max}. \tag{1.21}$$

### *1.3.2 Computation of the Dynamic Constraints*

By using the object-oriented approach described in [3], the aforementioned dynamic constraints can be easily computed at every state $[s,\dot{s}]^{\mathrm{T}}$.

Let the transmission of motion from the motion coordinate $s$ to the mass and force elements be given by a kinetostatic transmission element $\varphi_S$, denoted global kinematics. The concatenation of position, velocity, acceleration and force transmission functions of the global kinematics yields the inverse dynamics $\varphi_S^{D^{-1}}$ of the system, which maps the generalized coordinates and their time derivatives to a set of residual generalized forces

$$\underline{\bar{Q}} = \varphi_S^{D^{-1}}(s,\dot{s},\ddot{s}) = -\underline{m}(s)\,\ddot{s} - \underline{c}(s,\dot{s}) - \underline{d}(s), \tag{1.22}$$

at the input of the global kinematics. These residual forces can be used to generate $\underline{m}$, $\underline{c}$, and $\underline{d}$ of Eq. 1.13 at every configuration $s$ by the following simplified procedure:

(a) Computation of $\underline{d}$: Set, at the input of $\varphi_S^{D^{-1}}$, the generalized velocities to $\dot{s} = 0$ and the generalized accelerations to $\ddot{s} = 0$. Then, the terms $\underline{m}\,\ddot{s}$ and $\underline{c}$ of Eq. 1.13 vanish and the residual vector obtained at the input is exactly $-\underline{d}$.

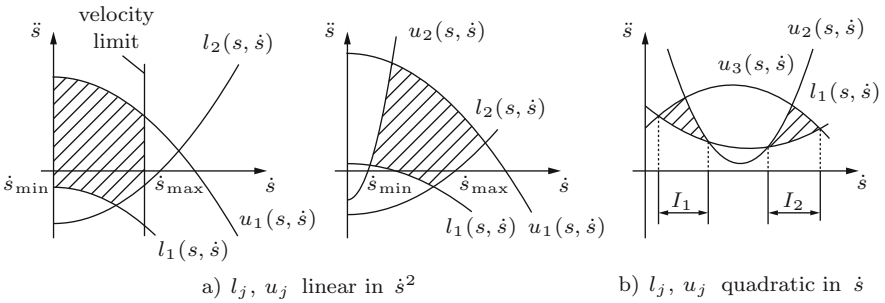



**Fig. 1.3** Admissible acceleration region for given configuration $s$

(b) Computation of $\underline{c}$: Eliminate the term $\underline{d}$ in the calculation of $\varphi_S^{D^{-1}}$ by 'switching off' the gravitational forces $\underline{Q}_G$, and set, at the input of $\varphi_S^{D^{-1}}$, the generalized velocities to $\dot{s} = 1$ and the generalized accelerations to $\ddot{s} = 0$. Then, the term $\underline{m}\,\ddot{s}$ of Eq. 1.13 vanishes and the residual vector obtained at the input is exactly $-\underline{c}$.
(c) Computation of $\underline{m}$: Similarly, eliminate the term $\underline{d}$ in the calculation of $\varphi_S^{D^{-1}}$ and set the input acceleration to $\ddot{s} = 1$. Then, the resulting force $\bar{Q}$ is exactly $-\underline{m}$.

The Jacobian matrices can be computed similarly, using only the force transmission functions, as proposed by [3]. Consider a kinetostatic element such as the one defined in Sect. 1.2.1. Setting all force components at the output of the transmission element besides the $j$th-one equal to zero, and the $j$th-one equal to one, yields a vector of generalized forces at the input the transmission element which is identical to the $j$th-column of the transposed Jacobian, thus to its $j$th-row.

### *1.3.3 Solution of the Time-Optimal Problem*

Equations 1.19 and 1.16 form a set of velocity and acceleration limits at every configuration $s$. These limits determine an admissible region in the plane $\dot{s} - \ddot{s}$ for each configuration $s$. Figure 1.3 shows typical admissible regions for two particular cases: (a) functions $l_j$, $u_j$ linear in $\dot{s}^2$, and (b) functions $l_j$, $u_j$ that are generally quadratic in $\dot{s}$. For the case of $l_j$, $u_j$ linear in $\dot{s}^2$ one can define one simple interval $[\dot{s}_{\min}, \dot{s}_{\max}]$ of admissible velocities for every configuration $s$. This assumption is not valid in general, since the admissible region could consist of not connected subregions leading to a set not connected admissible velocity intervals ($I_1$, $I_2$ in Fig. 1.3b).

The time optimal problem becomes particularly simple if all constraints are linear in $\dot{s}^2$. In this case, constraint equations of the form Eq. 1.15 can be written as

$$\hat{\underline{c}}_1(s)\dot{s}^2 + \hat{\underline{d}}_1(s) \leq \hat{\underline{m}}(s)\,\ddot{s} \leq \hat{\underline{c}}_2(s)\dot{s}^2 + \hat{\underline{d}}_2(s) \tag{1.23}$$

The function $g(s,\dot{s})$ is linear in $\dot{s}^2$, which means that the admissible regions in the plane $\dot{s}-\ddot{s}$ are simply connected and the set of admissible states has no holes in its interior. This allows for the definition of the functions $\dot{s}_{\min}(s)$ and $\dot{s}_{\max}(s)$ describing the maximally and minimally allowed admissible velocities $\dot{s}$ as a function of the motion coordinate $s$.

In many applications, the lower limiting curve $\dot{s}_{\min}(s)$ is required to be zero along the whole spatial path, so that all multibody system configurations are feasible at rest. However it actually does not need to be so in order for the time-optimal problem to have a feasible solution, as shown in Sect. 1.4.2.

The states that lie on the upper limiting curve $\dot{s}_{\max}(s)$ are classified in:

(a) Sinks, if $U(s,\dot{s}) = L(s,\dot{s}) > \dot{s}_{\max}\ \mathrm{d}\dot{s}_{\max}/\mathrm{d}s$
(b) Source, if $U(s,\dot{s}) = L(s,\dot{s}) < \dot{s}_{\max}\ \mathrm{d}\dot{s}_{\max}/\mathrm{d}s$
(c) Tangent points, elsewhere.

Moreover, the tangent points at which the velocity constraints described in Eq. 1.16 are active are called singular points, or singular arcs if they are connected.

The solution to the time optimal motion is a sequence of branches of maximal accelerations and maximal decelerations that lies in the feasible region and touches tangentially the upper limiting curve. At states $[s,\dot{s}]^{\mathrm{T}}$ that lie inside the feasible region, the solution consists of segments with maximal acceleration $U(s,\dot{s})$ and segments with maximal deceleration $L(s,\dot{s})$. At singular points, the extremal accelerations are further bounded by the upper limiting curve tangent $\mathrm{d}\dot{s}_{\max}/\mathrm{d}s$.

With these definitions, the following algorithm based on the one proposed by [6] has been constructed:

*Step 0*: Check if the initial state $s_0$ and the final state $s_{\mathrm{f}}$ are feasible for the given initial velocity $\dot{s}_0$ and final velocity $\dot{s}_{\mathrm{f}}$ respectively. If not, the problem has no feasible solution.

*Step 1*: Set a counter $k$ to 1. Integrate the equation $\ddot{s} = \max\{l_j(s,\dot{s})\}$ backwards in time from the final state $s = s_{\mathrm{f}}, \dot{s} = \dot{s}_{\mathrm{f}}$ until leaving the feasible region. Name the computed deceleration curve $\dot{s}^d(s)$.

*Step 2*: Integrate the equation $\ddot{s} = \min\{u_j(s,\dot{s})\}$ forwards in time from the initial state $s = s_0\ ,\ \ \dot{s} = \dot{s}_0$ until leaving the feasible region. Name the computed acceleration curve $\dot{s}^a_k(s)$. If the acceleration curve $\dot{s}^a_k(s)$ crosses the lower limiting curve $\dot{s}_{\min}(s)$, the problem is not feasible and the algorithm should be terminated. Else, continue.

*Step 3*: If $\dot{s}^a_k(s)$ crosses the deceleration curve $\dot{s}^d(s)$ terminate the algorithm: the intersection of both curves is the only switching point $S_k$. Otherwise, continue.

*Step 4*: Search forwards on the upper limiting curve $\dot{s}_{\max}(s)$ for the next tangency point $S_{k+1}$. The point $S_{k+1}$ is a switching point candidate.

*Step 5*: Integrate the equation $\ddot{s} = \max\{l_j(s,\dot{s}), \dot{s}_{\max}\mathrm{d}\dot{s}_{\max}/\mathrm{d}s\}$ backwards in time from the state $S_{k+1}$ until crossing one of the acceleration curves $\dot{s}^a_\ell(s)$, with $1 \le \ell \le k$. The intersection of both curves is the switching point $S_\ell$. Set $k = \ell$. Disregard the candidates $S_r$, with $r \le \ell$.

*Step 6*: Integrate the equation $\ddot{s} = \min\{u_j(s,\dot{s}), \dot{s}_{\max}\,\mathrm{d}\dot{s}_{\max}/\mathrm{d}s\}$ forward in time from the state $S_k$ until leaving the feasible region. Add one to the counter $k$. Name the

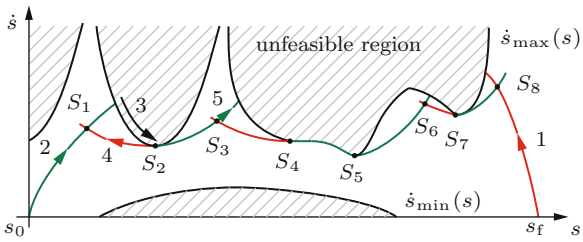



**Fig. 1.4** Time-optimal solution algorithm (no islands)

computed acceleration curve $\dot{s}_k^a(s)$. If the acceleration curve $\dot{s}_k^a(s)$ crosses the lower limiting curve $\dot{s}_{\min}(s)$, the problem is not feasible and the algorithm should be terminated. Else, go to step (3).

Figure 1.4 shows how a typical solution looks like.

For more general functions $l_j$, $u_j$, the feasible region may be composed of several unconnected regions of admissibility. In this case, the computation of the set of admissible states requires the solution of the scalar inequality Eq. 1.16, which involves the search for the zeros of $g(s, \dot{s})$ for given configurations $s$. This is possible only if enough information on the nature of the nonlinearities is available, as in the example presented in Sect. 1.4.1. If this is the case, all tangent points on the boundary of the admissible states can be computed. This allows for the construction of a directed graph containing a set of trajectories which connect the tangent points with the initial and final states, the highest of which yields the time-optimal motion.

## 1.4 Application Examples

### *1.4.1 Loading Cycles of Backhoe Excavators*

As a first example, consider the computation of the minimal time that an excavator would need to move along a typical load and haul path without violating the maximally allowed actuator forces as well as the maximally allowed overall hydraulic power consumption. A typical backhoe excavator consists of an under carriage, an upper carriage, a boom, a stick and a shovel (see Fig. 1.5). Its kinematical structure comprises three independent planar kinematical loops contained in the boom-stick plane as well as an independent rotation around a vertical axis.

The given path is interpolated in joint space using third order B-splines and it is assumed that the excavator starts its motion from and finishes it at rest. The dynamic constraints comprise the maximal hydraulic forces at the four actuators

$$Q_i^{\min} \leq \underline{Q} \leq Q_i^{\max}, \quad i = 1, 2, ..., 4, \tag{1.24}$$

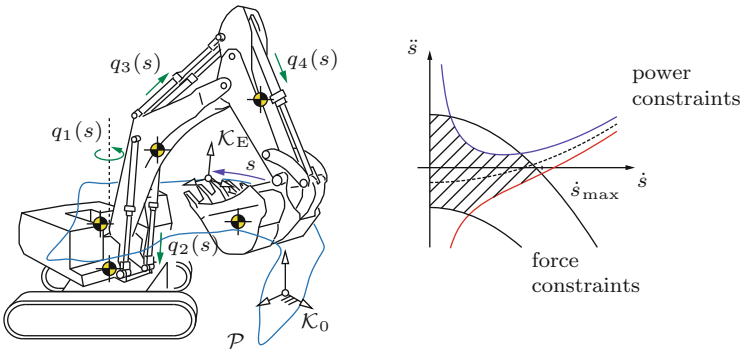



**Fig. 1.5** Loading cycle of a backhoe excavator

which leads to constraint equations of the form Eq. 1.19 with functions $l_j(s,\dot{s}), u_j(s,\dot{s})$ linear in $\dot{s}^2$, as well as the maximal hydraulic power

$$|\underline{\dot{q}}^{\mathrm{T}}\underline{Q}| \leq P_{\max}, \tag{1.25}$$

which together with Eqs. 1.9, 1.13, and 1.14 leads to

$$\frac{-P_{\max}}{\hat{m}(s)\dot{s}} - \frac{\hat{b}(s,\dot{s})}{\hat{m}(s)} \leq \ddot{s} \leq \frac{P_{\max}}{\hat{m}(s)\dot{s}} - \frac{\hat{b}(s,\dot{s})}{\hat{m}(s)}: \tag{1.26}$$

with

$$\hat{m} = {}_q\mathbf{J}_s^{\mathrm{T}}\mathbf{M}(s)\,{}_q\mathbf{J}_s > 0 \tag{1.27}$$

$$\hat{b} = {}_q\mathbf{J}_s^{\mathrm{T}}[\mathrm{M}(s)\,{}_q\mathbf{J}_s' + \bar{\underline{b}}(s)]\dot{s}^2 + {}_q\mathbf{J}_s{}^{\mathrm{T}}\underline{Q}_{\mathrm{G}}(s). \tag{1.28}$$

In this case, the function $g(s,\dot{s})$ of Eq. 1.16 has, for a given configuration $s$, at the most three zeros, all of which can be computed exactly. The representation of the acceleration constraints in the $\ddot{s}-\dot{s}$ plane is shown in Fig. 1.5. The force constraints form parabolas with vertices on the $\ddot{s}$ axis, whereas the power constraints approach hyperbolically to a similar parabola.

The time optimal solution is plotted in Fig. 1.6. The plots show alternating horizontal plateaus which are to be seen through the different curves, which indicates that at least one of the maximally allowed forces or the maximal power consumption is always used.

### *1.4.2 Waiter-Motion Problem for a Given Spatial Path*

As a second example, consider the so called "generalized waiter-motion problem". The well-known task consists in moving a tablet carrying a number of glasses from

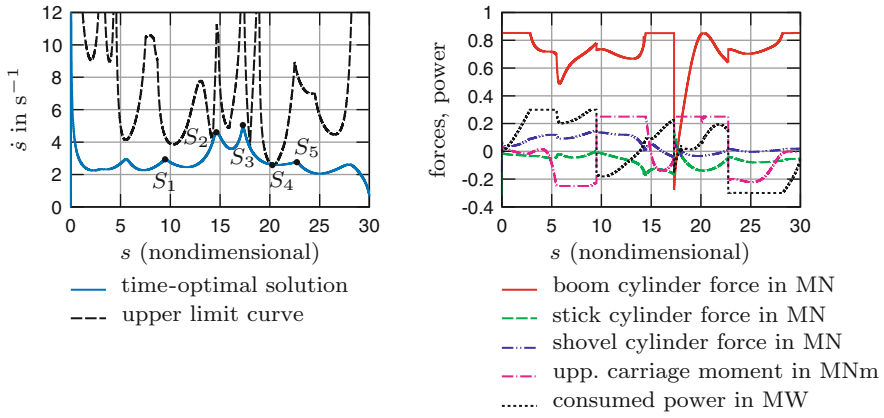



**Fig. 1.6** Optimized backhoe excavator loading cycle

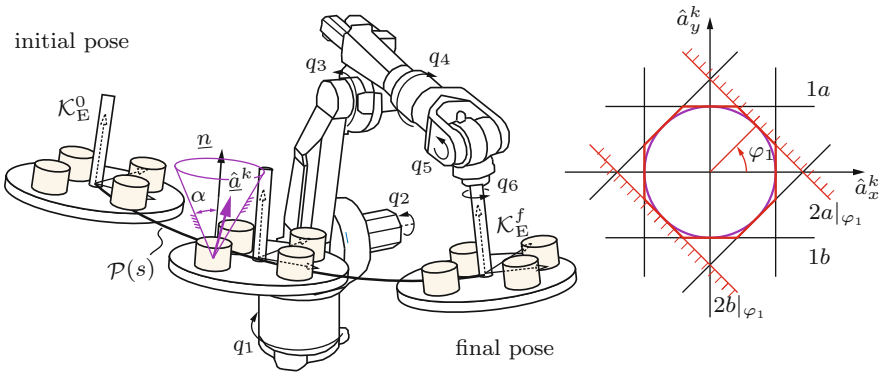



**Fig. 1.7** Generalized waiter-motion with no-sliding constraints

an initial pose $\mathcal{K}_{\mathrm{E}}^{0}$ to a final pose $\mathcal{K}_{\mathrm{E}}^{f}$ as fast as possible such that objects placed on it do not slide (see Fig. 1.7). The problem is solvable in closed form for the case of one single glass, but leads to a yet unsolved complex problem for the case of few glasses placed at different positions on the tablet.

The task considered here is the computation of the fastest motion along a prescribed spatial path given as a spatial interpolation of key poses of $\mathcal{K}_{\mathrm{E}}$. One set of constraints is given as limits in the joint velocities and accelerations in the form

$$\begin{aligned} \underline{\dot{q}}^{\min} \le \underline{\dot{q}} \le \underline{\dot{q}}^{\max} \\ \underline{\ddot{q}}^{\min} \le \underline{\ddot{q}} \le \underline{\ddot{q}}^{\max} \end{aligned} \tag{1.29}$$

with $\dot{q}_i^{\min}$, $\dot{q}_i^{\max}$, $\ddot{q}_i^{\min}$ and $\ddot{q}_i^{\min}$ constant. These values are typically provided by the manufacturer and are usually pre-programmed in the robot controller as soft-limits.

A second set of constraints is given by the sticking ('no sliding') condition for every object $k$ on the tablet

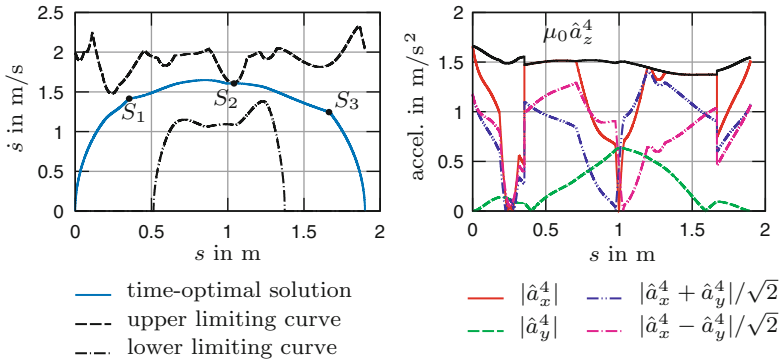



**Fig. 1.8** Optimized waiter-motion along given trajectory

$$\frac{\hat{\underline{a}}^k \cdot \underline{n}}{\|\hat{\underline{a}}^k\|_2} \geq \cos\alpha, \quad \text{with } \hat{\underline{a}}^k = \underline{a}^k + \underline{g}, \tag{1.30}$$

where $\underline{a}^k$ is the acceleration of object $k$ and $\underline{g}$ is the gravity vector, $\underline{n}$ is the normal vector of the tablet plane, and $\mu_0 = \tan\alpha$ is the dry friction coefficient between plane and objects. These $k$ additional dynamic constraints can be rewritten as

$$\sqrt{[\hat{a}_x^k]^2 + [\hat{a}_y^k]^2} \leq \mu_0 \hat{a}_z^k, \quad \text{with } \hat{\underline{a}}^k = {}_k\mathbf{J}_s\,\ddot{s} + {}_k\mathbf{J}_s'\,\dot{s}^2 - \mathbf{R}_k^{\mathrm{T}}\underline{g}, \tag{1.31}$$

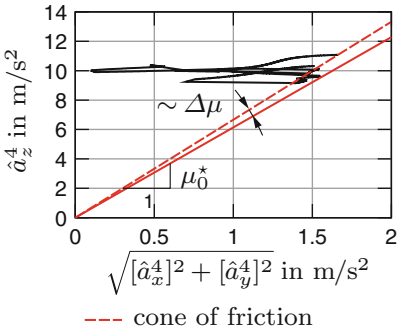
where ${}_k\mathbf{J}_s$ is the Jacobian mapping the linear velocities $\dot{s}$ along the spatial path to the velocities of object $k$, and $\mathbf{R}_k$ is the transformation matrix from the inertial frame to the local coordinate frame of object $k$.

Clearly, equations of the form Eq. 1.31 are nonlinear in the unknowns $\ddot{s}$, which makes their treatment with the previous methods infeasible. However, it is possible to approximate these constraints by replacing the friction cone by a friction polyhedra (Fig. 1.7) given by the equations

$$\begin{aligned} &1) \qquad |\hat{a}_x^k| \leq \mu_0 \hat{a}_z^k \\ &2) \quad \left| -\frac{\hat{a}_x^k}{\tan(\varphi_i)} + \hat{a}_y^k \right| \leq \frac{\mu_0 \hat{a}_z^k}{\sin(\varphi_i)}, \end{aligned} \tag{1.32}$$

defined by the discretization angles $\varphi_i = \frac{i\pi}{2^{p-1}}$, with $i = 1, 2, ..., 2^{p-1} - 1$, for each $\hat{a}_z^k$. Note that this approximation can be arbitrarily refined by choosing a sufficiently large integer $p$.

Equation 1.32 together with Eq. 1.29 form a system of constraints that are linear in $\dot{s}^2$, so that simply connected admissible acceleration regions are granted. The solution for the case of four objects symmetrically distributed on the tablet and a friction pyramid approximation with $p = 4$ is shown in Fig. 1.8. The plot at the right shows the accelerations of the fourth glass ($k = 4$).




| $p$ | $2^{p-1}$ | $\frac{\Delta\mu}{\mu_0}$ (%) | $t_f$ (s) | CPU (s) |
|---|---|---|---|---|
| 3 | 4 | 5.851 | 1.7000 | 5.906 |
| 4 | 8 | 5.850 | 1.7004 | 7.361 |
| 5 | 16 | 1.958 | 1.7033 | 11.230 |
| 6 | 32 | 0.483 | 1.7043 | 23.219 |
| 7 | 64 | 0.121 | 1.7046 | 65.183 |

**Fig. 1.9** Delimitation of violations by relaxing the friction constraints

The small violations of the sticking condition are a consequence of the pyramid of friction approximation. They can be quantified by defining the friction correction $\Delta\mu$ (see Fig. 1.9), which is the smallest increase of the actual friction coefficient $\mu_0$ that makes the computed motion feasible. As shown by the numerical experiments shown in Fig. 1.9, by increasing the integer $p$ the accuracy of the constraint equations can be increased arbitrarily, although at expense of exponentially increasing CPU time.

## 1.5 Conclusions

This work extends classical fixed-geometry time-optimal path planning methods by (1) allowing for the consideration of sticking conditions and overall power consumption constraints, and (2) the allowing for general paths featuring smooth interpolation of angular acceleration as well as arbitrary multibody systems comprising multiple loops. The paper shows how the concept of motion interpolation as a kinetostatic transmission element allows for an efficient method to generate the transmission equations that lead to the formulation of the solution algorithm. The constraints can be straightforwardly constructed in terms of velocities, accelerations and forces at any place along the kinematical skeleton. The practical applicability of these extensions is illustrated by two applications from robotics and mining industry.

## References

1. Bobrow JE, Dubowsky S, Gibson JS (1985) Time-optimal control of robotic manipulators along specified paths. Int J Robot Res 4(3):3–17
2. Dierckx P (1993) Curve and surface fitting with splines. Clarendon Press, Oxford
3. Kecskem'ethy A, Hiller M (1994) An object-oriented approach for an effective formulation of multibody dynamics. Comput Method Appl Mech Eng 115:287–314

4. Messner L, Gattringer H, Bremer H (2011) Generating speed, torque, and jerk limited trajectories along specified geometric paths in realtime. In: Proceedings of ROBO 2011, Pitsburgh
5. Pfeiffer F, Johanni R (1987) A concept for manipulator trajectory planning. IEEE J Robot Autom 3:115–123
6. Shiller Z, Lu HH (1990) Robust computation of path constrained time optimal motions. Proc IEEE Int Conf Robot Autom 1:144–149, Cincinnati
7. Shin KG, McKay ND (1985) Minimum-time control of robotic manipulators with geometric path constraints. IEEE Trans Autom Control 30(6):531–541
8. Tändl M, Kecskem'ethy A, Schneider M (2007) A design environment for industrial roller coasters. In: CD Proceedings of the ECCOMAS thematic conference on advances in computational multibody dynamics, Milano
9. Verscheure D, Demeulenaere B, Swevers J, De Schutter J, Diehl M (2009) Timeoptimal path tracking for robots: a convex optimization approach. IEEE Trans Autom Control 54 (10):2318–2327

# Chapter 2
# Efficient Online Computation of Smooth Trajectories Along Geometric Paths for Robotic Manipulators

**Lukas Messner, Hubert Gattringer, and Hartmut Bremer**

**Abstract** This paper presents a fast computation method for time-optimal robot state trajectories along specified geometric paths. A main feature of this new algorithm is that joint positions can be generated in realtime. Hence, not only joint velocities and accelerations limits but also constraints on joint jerks and motor torques can be considered. Jerk limits are essential to avoid vibrations due to (not-modeled) gear or structure flexibilities. For the limitation of motor torques a complete dynamic robot model including Coulomb and viscous friction is used. The underlying optimal control problem is found by projecting the problem onto the geometric path. The resulting state vector contains path position, speed and acceleration while path jerk is used as input. From optimal control theory it follows that the path jerk has to be chosen at its boundaries, which can be computed for each state in each step. Continuous state progress is assured via so called test trajectories which are additionally computed in each step. As an example the algorithm is applied to a six-axis industrial robot moving along a straight line in Cartesian space.

## 2.1 Introduction

Despite the fact that in the last decades big effort has been put into the development of path planning techniques, most of these methods are not used in industry and the capabilities of robots are often not fully exploited. One reason for this is, that most algorithms are designed for an offline computation of a desired trajectory for the robot's joints $\mathbf{q}_d(t)$ although an online computation is desirable for mainly two reasons: (1) A movement described with a robotic program should start without computation delays and (2) a flexible production system should be able to react on unforeseeable events in realtime.

L. Messner (✉) • H. Gattringer • H. Bremer
Johannes Kepler University, Altenbergerstr. 69, Linz 4040, Austria
e-mail: lukas.messner@jku.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_2, 

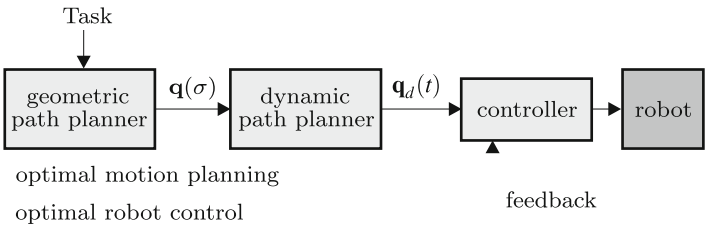



**Fig. 2.1** The decoupled approach. A geometric path $\mathbf{q}(\sigma)$ from a geometric path planner is the basis for a time parametrization in the dynamic path planner. The feedback controller is then tracking so computed reference trajectories $\mathbf{q}_d(t)$

Typically, the general online path planning problem leads to a realtime optimal control problem [5]. Unfortunately, current industrial realtime environments are not powerful enough to solve the general problem (e.g. for a six axis robot) using state of the art techniques, like model predictive control (MPC) [9]. However, in this paper a typical industrial case with a predefined geometric path $\mathbf{q}(\sigma)$ is considered (e.g. defined by a robotic program). Therefore, it is known that the optimal control problem can be greatly reduced by projecting it onto the path parameter $\sigma$. This separation of the geometric path planning problem, including the inverse kinematics, and the dynamic path planning problem for finding a course in time $\sigma(t)$ was firstly introduced by Bobrow et al. [3] and is called *decoupled approach* [15]. The resulting control setup is also depicted in Fig. 2.1.

There exist many well studied methods for solving the optimal control problem arising from the decoupled approach which can be divided into dynamic programming [2, 11], direct solution methods [8, 13, 18] and indirect solution methods [3, 17]. However, only few existing methods are designed for realtime usage and they often do not fulfill all industrial demands. For example vibration avoiding smoothing techniques like jerk limitations are not considered in Pardo-Castellote's approach [15]. Other methods do not consider a dynamic robot model for the consideration of actuator torques (e.g. [7] or [1]) or they are limited to special geometric paths like straight lines [14].

This paper presents an algorithm which overcomes the realtime problem by not solving the optimal control problem for the whole path at once, but only for a short distance in each step. Firstly, the problem is projected onto the geometric path by introducing a path position, speed and acceleration state vector (Sect. 2.2). With path jerk as input, it is shown that desired joint limits can be formulated as pure state or mixed input state inequality conditions for that system. In Sect. 2.3, possible input values are derived from Pontryagin's Maximum Principle [16]. Thereafter, a time discretization is introduced and continuable states are defined. Finally, in Sect. 2.4 test trajectories are used to check for continuable states. These trajectories

form the key computation for the presented discrete online algorithm. To demonstrate the performance of the method, results for a six-axis industrial robot are shown in Sect. 2.5.

## 2.2 Problem Statement

In the following, a geometric path

$$\mathbf{q}(\sigma) = [q_1(\sigma), \ldots, q_n(\sigma)]^{\mathrm{T}} \qquad \sigma \in [\sigma_0, \sigma_e] \tag{2.1}$$

is given for a fully actuated robotic manipulator with $n$ joints $q_1, \ldots, q_n$. It is assumed that $\mathbf{q}(\sigma)$ is at least three times differentiable with respect to the arbitrarily chosen path parameter $\sigma$.

For the limitation of actuator torques or forces $\boldsymbol{\tau}$, a nonlinear dynamic model is used in the general form

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{D}\,\dot{\mathbf{q}} + \mathbf{D}_c\,\mathrm{sign}(\dot{\mathbf{q}}) \tag{2.2}$$

with the mass matrix $\mathbf{M}$, the Coriolis and centrifugal force matrix $\mathbf{C}$, which is linear in $\dot{\mathbf{q}}$ and the gravity force vector $\mathbf{G}$. Viscous friction is considered with $\mathbf{D}\,\dot{\mathbf{q}}$ and $\mathbf{D}_c\,\mathrm{sign}(\dot{\mathbf{q}})$ are Coulomb friction torques and forces [4].

### *2.2.1 Optimal Control Problem*

For a given geometric path (2.1) the whole path planning problem is reduced to the task of finding a function $\sigma(t)$. It will be shown that constraints and an optimization criterion for this function lead to an optimal control formulation in the form

$$\min_{u(t)} J := \int_0^{t_e} l(\mathbf{x}(t), u(t))\mathrm{d}t \tag{2.3}$$

$$\text{subject to } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u),\ \mathbf{x}(0) = \mathbf{x}_0,\ \mathbf{x}(t_e) = \mathbf{x}_e, \tag{2.4}$$

$$\left.\begin{aligned} \mathbf{h}(\mathbf{x}) &\le 0 \\ \mathbf{g}(\mathbf{x}, u) &\le 0 \end{aligned}\right\} \ t \in [0, t_e], \tag{2.5}$$

where $J$ is a cost functional, $\mathbf{f}$ denotes a dynamic system and $\mathbf{h}$ and $\mathbf{g}$ consider pure state and mixed input-state inequality conditions [5].

Herein, the pure time optimal case will be considered by choosing $l(\mathbf{x}(t), u(t)) = 1$. Furthermore, limitations on joint speeds, accelerations, jerks and motor torques will be taken into account with

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \dot{\mathbf{q}}(x_1, x_2) - \dot{\mathbf{q}}_{max} \\ -\dot{\mathbf{q}}(x_1, x_2) + \dot{\mathbf{q}}_{min} \\ \ddot{\mathbf{q}}(x_1, x_2, x_3) - \ddot{\mathbf{q}}_{max} \\ -\ddot{\mathbf{q}}(x_1, x_2, x_3) + \ddot{\mathbf{q}}_{min} \\ \boldsymbol{\tau}(x_1, x_2, x_3) - \boldsymbol{\tau}_{max} \\ -\boldsymbol{\tau}(x_1, x_2, x_3) + \boldsymbol{\tau}_{min} \end{bmatrix}, \tag{2.6}$$

$$\mathbf{g}(\mathbf{x}, u) = \begin{bmatrix} \dddot{\mathbf{q}}(x_1, x_2, x_3, u) - \mathbf{j}_{max} \\ -\dddot{\mathbf{q}}(x_1, x_2, x_3, u) + \mathbf{j}_{min} \end{bmatrix}, \tag{2.7}$$

by selecting path position, speed and acceleration as components of the state vector $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}} = [\sigma, \dot{\sigma}, \ddot{\sigma}]^{\mathrm{T}}$ and $u = \dddot{\sigma}$ as input of the dynamic system $\mathbf{f}(\mathbf{x}, u) = [x_2, x_3, u]^{\mathrm{T}}$. Dependencies on system input and state components in (2.6), and (2.7) can be easily seen by expanding the time derivatives

$$\dot{\mathbf{q}}(x_1, x_2) = \mathbf{q}'(x_1)\, x_2 \tag{2.8}$$

$$\ddot{\mathbf{q}}(x_1, x_2, x_3) = \mathbf{q}''(x_1)\, x_2^2 + \mathbf{q}'(x_1)\, x_3 \tag{2.9}$$

$$\dddot{\mathbf{q}}(x_1, x_2, x_3, u) = \mathbf{q}'''(x_1)\, x_2^3 + \mathbf{q}''(x_1) 3 x_2 x_3 + \mathbf{q}'(x_1)\, u\,, \tag{2.10}$$

using the differential operator $(.)' := \frac{\partial(.)}{\partial \sigma}$. For the actuator torques and forces, the dynamic equation (2.2) is rewritten to

$$\boldsymbol{\tau}(x_1, x_2, x_3) = \mathbf{a}_0(x_1) + \mathbf{a}_1(x_1)\, x_2 + \mathbf{a}_2(x_1)\, x_2^2 + \mathbf{a}_3(x_1)\, x_3\,, \tag{2.11}$$

with purely $\sigma$-depending parameters

$$\begin{aligned} \mathbf{a}_0(x_1) &= \mathbf{G}(\mathbf{q}(x_1)) + \mathbf{D}_c\, \mathrm{sign}(\mathbf{q}'(x_1)) \\ \mathbf{a}_1(x_1) &= \mathbf{D}\, \mathbf{q}'(x_1) \\ \mathbf{a}_2(x_1) &= \mathbf{M}(\mathbf{q}(x_1))\, \mathbf{q}''(x_1) + \mathbf{C}(\mathbf{q}(x_1), \mathbf{q}'(x_1))\, \mathbf{q}'(x_1) \\ \mathbf{a}_3(x_1) &= \mathbf{M}(\mathbf{q}(x_1))\, \mathbf{q}'(x_1)\,. \end{aligned}$$

For an efficient computation of the inequality conditions (2.6), and (2.7) the existence of purely $\sigma$-depending parameters is advantageous because a preparation (e.g. using a spline approximation) is possible in advance.

## 2.3 Optimal System Inputs

For solving an optimal control problem (2.3), (2.4), and (2.5) various techniques exist [5]. Similar to indirect solution methods optimality conditions derived from Pontryagin's Maximum Principle give useful information for the choice of optimal system inputs. Therefore, a Hamiltonian is introduced as

$$H(\mathbf{x}, u, \lambda) := \lambda_0 l + \lambda^{\mathrm{T}} \mathbf{f}(\mathbf{x}, u) = \lambda_0 + \lambda_1 x_2 + \lambda_2 x_3 + \lambda_3 u\,, \tag{2.12}$$

for the problem (2.3), (2.4), and (2.5) with Lagrangian Multipliers $\lambda_0, \ldots, \lambda_3$ [5].

For an optimal solution $(\mathbf{x}^*, u^*, \lambda^*)$ of a problem (2.3), (2.4), and (2.5) the first necessary condition is given by [6]

$$H(\mathbf{x}^*, u^*, \lambda^*) \leq H(\mathbf{x}^*, u, \lambda^*)\,. \tag{2.13}$$

This means that for an optimal point $(\mathbf{x}^*, \lambda^*)$ on the trajectory, the optimal input $u = u^*$ has to minimize the Hamiltonian $H(\mathbf{x}^*, u, \lambda^*)$. Hence, the optimality condition (2.13) for the Hamiltonian (2.12) can be rewritten to

$$u^* = \begin{cases} u_{max}(\mathbf{x}^*), & \text{if } \lambda_3^* < 0 \\ u_{min}(\mathbf{x}^*), & \text{if } \lambda_3^* > 0 \\ \text{undefined}, & \text{if } \lambda_3^* = 0 \end{cases}, \tag{2.14}$$

assuming upper and lower bounds $u_{max}(\mathbf{x})$ and $u_{min}(\mathbf{x})$ for the input $u$ can be defined according to the inequality conditions (2.5) and the system equation (2.4). Additional optimality conditions could give more information about the Lagrangian Multiplier $\lambda_3$, but it is known that pure state constraints lead to complex switching point analysis [5]. However, assuming no singular arcs [5] exist with $\lambda_3^* = 0$ on a non empty time interval $(\tau_1, \tau_2) \subset [t_0, t_e]$, the information given by the optimality condition (2.14) is already very useful because it means that $u^*$ can only be either $u_{max}(\mathbf{x})$ or $u_{min}(\mathbf{x})$.

### *2.3.1 Discretization*

In this section a discretization of the original problem (2.3), (2.4), and (2.5) is used to subsequently define a discrete online algorithm.

With sample times $t_k = \sum_{i=0}^{k} T_i, k = 0, \ldots, n$, a piecewise constant input $u(t) = u_k$, $t \in [t_k, t_{k+1}]$ and discrete states $\mathbf{x}_k = \mathbf{x}(t_k)$, a time discretization of the system equation (2.4) is given by [10]

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{b}_k u_k \,,$$
$$\mathbf{A}_k = \begin{bmatrix} 1 & T_k & \frac{T_k^2}{2} \\ 0 & 1 & T_k \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_k = \begin{bmatrix} \frac{T_k^3}{6} \\ \frac{T_k^2}{2} \\ T_k \end{bmatrix}. \tag{2.15}$$

Inequality conditions (2.5) can be written as

$$\begin{matrix} \mathbf{g}(\mathbf{x}_k, u_k) \leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}_k) \leq \mathbf{0} \end{matrix}, k = 0, \ldots, n \,, \tag{2.16}$$

which means that the original inequality conditions (2.5) must hold at discrete time steps $t_k$, $k = 0, \ldots, n$, but not in between. This simplification is common for many methods and in practice small enough sample times $T_k$ deliver satisfying results.

### 2.3.2 Input Bounds

In this section a computation rule for the input bounds $u_{max}(\mathbf{x})$ and $u_{min}(\mathbf{x})$, defined in the beginning of Sect. 2.3, shall be found.

The set of allowed input values $u_k$ is clearly limited by the mixed input-state constraints $\mathbf{g}(\mathbf{x}, u)$. In the discrete case pure state constraints $\mathbf{h}(\mathbf{x})$ can be taken into account by considering the next step $\mathbf{h}(\mathbf{x}_{k+1}(\mathbf{x}_k, u_k))$ where $u_k$ automatically appears. To avoid time consuming computations for a lower and upper bound of a so defined set

$$S_k = \{u_k \in \mathbb{R},\, \mathbf{g}(\mathbf{x}_k, u_k) \leq \mathbf{0},\, \mathbf{h}(\mathbf{x}_{k+1}(\mathbf{x}_k, u_k)) \leq \mathbf{0}\} \,,$$

it is useful to introduce a $u_k$ independent Taylor series estimation of the next state $\tilde{\mathbf{x}} = \mathbf{A}_k \mathbf{x}_k + \mathbf{b}_k 0$. Together with inequality conditions (2.6), and (2.7), this approximation lead to a set of linear inequality conditions for $u_k$, namely

$$\tilde{S}_k = \left\{u_k \in \mathbb{R},\ \boldsymbol{\beta}_{k,min} \leq \boldsymbol{\alpha}_k\, u_k \leq \boldsymbol{\beta}_{k,max}\right\}$$

with

$$\boldsymbol{\alpha}_k = \begin{bmatrix} \mathbf{q}' \\ \tilde{\mathbf{a}}_3\, T_s \\ \tilde{\mathbf{q}}' T_s \\ \tilde{\mathbf{q}}' \frac{T_s^2}{2} \end{bmatrix}, \quad \boldsymbol{\beta}_{k,\,\substack{min \\ max}} = \begin{bmatrix} \mathbf{j}_{\substack{min \\ max}} - (\mathbf{q}''' x_{2,k}^3 + \mathbf{q}'' 3 x_{2,k} x_{3,k}) \\ \boldsymbol{\tau}_{\substack{min \\ max}} - (\tilde{\mathbf{a}}_0 + \tilde{\mathbf{a}}_1\, \tilde{x}_2 + \tilde{\mathbf{a}}_2\, \tilde{x}_2^2 + \tilde{\mathbf{a}}_3\, x_{3,k}) \\ \ddot{\mathbf{q}}_{\substack{min \\ max}} - (\tilde{\mathbf{q}}'' x_{2,k}^2 + \tilde{\mathbf{q}}' x_{3,k}) \\ \dot{\mathbf{q}}_{\substack{min \\ max}} - \tilde{\mathbf{q}}' (x_{2,k} + x_{3,k} T_s) \end{bmatrix}$$

and $\tilde{(.)}$ denoting values computed with the estimated state $\tilde{\mathbf{x}}$. This directly gives a simply to compute estimation for $u_{k,max}$ and $u_{k,min}$ with $u_{k,min} \approx \inf\{\tilde{S}_k\}$ and $u_{k,max} \approx \sup\{\tilde{S}_k\}$.

### 2.3.3 Not Allowed States

In addition to the rule, how to compute $u_{k,max}$ and $u_{k,min}$ (see Sect. 2.3.2) it has to be clarified when to use $u_{k,max}$ and when $u_{k,min}$. When looking at the objective function, which can be rewritten as

$$J = \sum_{k=0}^{n-1} T_k = \sum_{k=0}^{n-1} \int_{x_{1,k}}^{x_{1,k+1}} \frac{1}{x_2} \mathrm{d}x_1, \tag{2.17}$$

choosing $u_{k,max}$ would be obviously the best to maximize the path speed $x_2$.

Additionally, it has to be considered that a sequence $\mathbf{x}_k, u_k \in S_k,\ k = j, j+1, \ldots,$ can potentially end in an empty set $S_k = \{\}$, which would mean that a not allowed state is reached and one of the inequality conditions (2.16) is violated.

States $\mathbf{x}_j$ belonging to a special subset $N \subset \mathbb{R}^3$ which can be continued with a sequence $u_k, k = j, j+1, \ldots,$ without reaching a not allowed state shall be called continuable states. Clearly, only continuable states should be considered but it is difficult to find a simple condition for $\mathbf{x}_j \in N$ due to the infinite horizon of the previous condition. However, if a finite trajectory $\mathbf{x}_k, u_k \in S_k, k = j, \ldots, m-1,$ which ends in a rest position $\mathbf{x}_m = [x_{1,m}\ \ 0\ \ 0]^{\mathrm{T}}$ can be found, it means that with $u_k = 0$, $\mathbf{x}_k = \mathbf{x}_m$ and $S_k = S_{m-1} \neq \{\}$ for all $k \geq m$ this rest position can be continued and it is a sufficient condition for a continuable state $\mathbf{x}_j$. A method for computing such short trajectories for testing if a state is continuable or not, is introduced in the subsequent section.

## 2.4 A Discrete Online Algorithm

In the previous Sect. 2.3 conditions for optimal system inputs $u_k$ are found. Based on those conditions, the idea for the online algorithm is to test in each step $u_{k,max}$ as input and to check if the resulting next state is continuable. Therefore, in the subsequent section, so called test trajectories are introduced.

### 2.4.1 Test Trajectories

As described in Sect. 2.3.3, if a test trajectory $\xi = (\bar{\mathbf{x}}_0, \ldots, \bar{\mathbf{x}}_m, \bar{u}_0, \ldots, \bar{u}_m)$ with a given start state $\bar{\mathbf{x}}_0$ and a desired rest position $\bar{\mathbf{x}}_{m,d} = [\sigma_{r,d}\ \ 0\ \ 0]^{\mathrm{T}}$ can be found, it means that all states $\bar{\mathbf{x}}_j,\ j = 0, \ldots, m$ of such a test trajectory are continuable states. Choosing the input at the boundaries in three time intervals $\tau_1, \tau_2$ and $\tau_3$ in the form

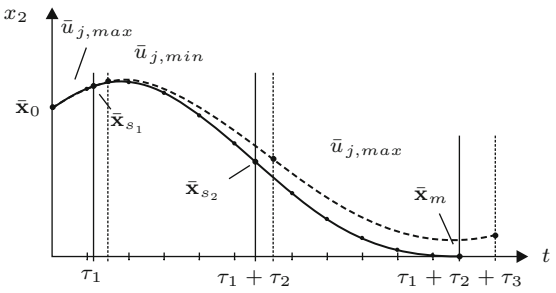



**Fig. 2.2** Test trajectory for an initial state $\bar{\mathbf{x}}_0$. The solution of the 2PBVP is depicted with the solid line. Inappropriately chosen time intervals $\tau_1, \tau_2$ and $\tau_3$ lead to a final state which is not a rest position (*dashed line*)

$$\bar{u}_j = \begin{cases} \bar{u}_{j,max} & \text{if } t_j \in [0, \tau_1) \\ \bar{u}_{j,min} & \text{if } t_j \in [\tau_1, \tau_1 + \tau_2) \\ \bar{u}_{j,max} & \text{if } t_j \in [\tau_1 + \tau_2, \tau_1 + \tau_2 + \tau_3) \end{cases} \tag{2.18}$$

result in a trajectory with three degrees of freedom $\tau_1$, $\tau_2$ and $\tau_3$ (Fig. 2.2). Finding correct values for $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^{\mathrm{T}}$ to end in the desired rest position $\bar{\mathbf{x}}_{m,d}$ is a two point boundary value problem (2PBVP). Therefore, known methods like single shooting [5] can be applied. In case of a free rest position $\sigma_{r,d}$ and fixed interval time $\tau_1$, the 2PBVP reduces to a unknown $\bar{\boldsymbol{\tau}} = [\tau_2, \tau_3]^{\mathrm{T}}$.

It is clear that such a test trajectory does not exist for a not continuable state $\bar{\mathbf{x}}_0$ (see Sect. 2.3.3). But if it exists, all states $\bar{\mathbf{x}}_0, \ldots, \bar{\mathbf{x}}_m$ are known to be continuable. This test for continuable states is the basis for the online algorithm defined in the following section.

### 2.4.2 The Algorithm

Now the idea for an algorithm solving the optimal control problem (2.3), (2.4), and (2.5) is to use a maximum path jerk $u_{k,max}$ in each step as input for the discrete system (2.15) except the test trajectory computation fails. A detailed algorithm is described in the following.

1. Start with $\mathbf{x}_0 = [\,\sigma_0 \;\; 0 \;\; 0\,]^{\mathrm{T}}$ and $k = 0$.
2. Try to compute a test trajectory $\xi_k = (\bar{\mathbf{x}}_{k,0}, \ldots, \bar{\mathbf{x}}_{k,m_k}, \bar{u}_{k,0}, \ldots, \bar{u}_{k,m_k})$ with $\bar{\mathbf{x}}_{k,0} = \mathbf{x}_k$, fixed interval $\tau_1 = T_s$ and a free end position $\sigma_{r,d}$.
3. If the test trajectory $\xi_k$ exceeds the end position $\sigma_e$, recompute the test trajectory with fixed end position $\bar{\mathbf{x}}_{k,m_k,d} = [\,\sigma_e \;\; 0 \;\; 0\,]^{\mathrm{T}}$ and free interval length $\tau_1$. If successful, go to 7.
4. If $\xi_k$ was not computed successfully, take the previous test trajectory but without the first state, thus

$$\xi_k = (\bar{\mathbf{x}}_{k-1,1}, \ldots, \bar{\mathbf{x}}_{k-1,m_{k-1}}, \bar{u}_{k-1,1}, \ldots, \bar{u}_{k-1,m_{k-1}}).$$

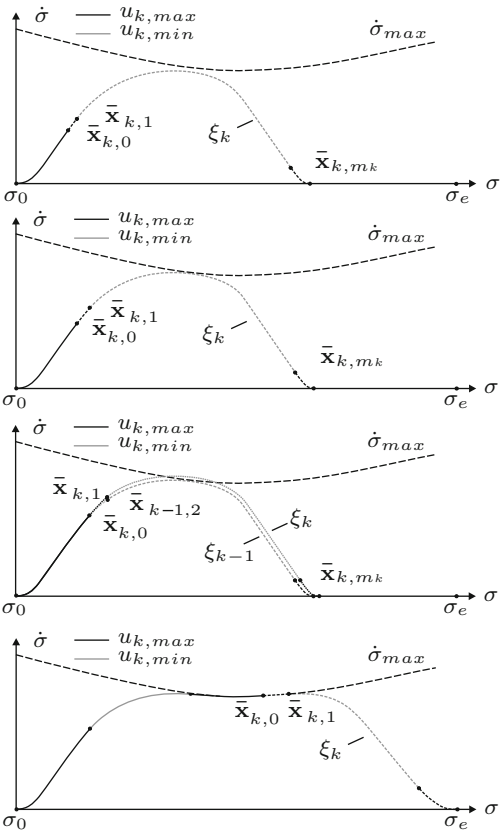



**Fig. 2.3** Three consecutive iterations and the last step of the algorithm. The presence of dynamic limits is represented by the *dashed line* $\dot{\sigma}_{max}$. In the first two steps it can be seen that a test trajectory is computed successfully without violating the limit. Therefore, the state $\bar{\mathbf{x}}_{k,1}$ is continuable and can be outputted. In the third step this is not the case and a state from the previous test trajectory $\bar{\mathbf{x}}_{k-1,2}$ is taken instead. The figure on the bottom shows the last step in which the test trajectory reaches the end of the path

5. Output $\mathbf{x}_k = \bar{\mathbf{x}}_{k,0}$, $u_k = \bar{u}_{k,0}$ and choose $\mathbf{x}_{k+1} = \bar{\mathbf{x}}_{k,1}$.
6. Increase $k$ by one and go to 2.
7. Output the whole test trajectory until the end of the path $\mathbf{x}_{k+j} = \bar{\mathbf{x}}_{k,j}, u_k = \bar{u}_{k,j}$ for $j = 0, \ldots, m_k$.

The evolution of the algorithm is also shown in Fig. 2.3. An iteration step of the algorithm always includes a test trajectory computation. Therefore, it is useful to know that, due to small iteration steps, time intervals $\tau_2$ and $\tau_3$ from a previous step $k-1$ can be assumed to be good starting values for the 2PBVP in the actual step $k$.

In each iteration step, a path jerk $u_{k,max}$ is used as long as test trajectories are computed successfully. If the computation fails, the algorithm stays on the last successfully computed test trajectory until a new test trajectory is found. Therefore, $u_{k,max}$ is used whenever possible, otherwise $u_{k,min}$ is taken as input for the discrete system (2.15). This means that the necessary condition for optimality (Sect. 2.3) is fulfilled.

### 2.4.3 Realtime Capability

The fact that each iteration step of the algorithm presented in Sect. 2.4.2 outputs a new path position $\sigma_k = x_{1,k}$ (and therefore also joint positions $\mathbf{q}(\sigma_k)$) makes this algorithm capable for realtime usage. Therefore, it is important to guarantee a maximum computation time for each step which is smaller than the chosen cycle time $T_s$. This is possible if the maximum computation time for a single test trajectory is limited by choosing a maximum trajectory length and a limitation of iterations for the 2PBVP. Exceeding these limits can be treated like a not successful test trajectory computation in the algorithm.

Experiments show, that for typical robotic applications (see Sect. 2.5) small enough sample times can be reached. Even smaller sample times, potentially needed for feedback controller, can be realized by simply resampling the result using (2.15), see [10]. Additionally, feed forward strategies [12] can be easily provided with set speeds, accelerations and torques resulting from (2.8), (2.9) and (2.11).

Due to the fact that the last point of each test trajectory $\bar{\mathbf{x}}_{k,m_k}$ is at the same time the most advanced point on the path, it is possible to change any information beyond that point like speed limits or the geometric path $\mathbf{q}(\sigma)$, $\sigma > \bar{x}_{k,m_k}$, without influencing the result. This means, that a geometric path can be prepared in the background while the algorithm is running and features like online speed adjustment (override) can be easily realized. In practice, it is also advantageous that in each time step a test trajectory is present which can bring the robot as fast as possible to a stop if required (e.g. emergency stop).

## 2.5 Example

In this section the presented algorithm is applied to a six-axis robot controlled by a standard industrial PC with a 1.4 GHz processor. For limiting motor torques, a full dynamic model with identified parameters is used. Velocity limits are taken from manufacturer's data sheet, acceleration limits are not needed (due to torque limitations) and jerk limits are derived from experimental results such that vibrations are extensively avoided.

A C-implementation of the algorithm for the standard controller reveals possible sample times smaller than $T_s = 2.4$ ms. This means that the realtime condition for the computation time of each iterations step is never violated.

In the following, a MATLAB-implementation of the presented algorithm is used to be able to compare the results with other state of the art (offline) methods. As an example, a geometric path is computed from an inverse kinematic transformation of a straight line in Cartesian space with fixed orientation of the tool center point (see Fig. 2.4). The resulting speed profile $\dot{\sigma}(t)$ and all consecutive test trajectories computed with a sample time $T_s = 10$ ms are depicted in Fig. 2.5.

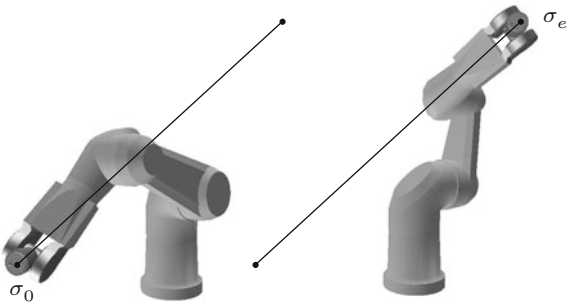



**Fig. 2.4** A geometric path for a six-axis robot moving along a straight line

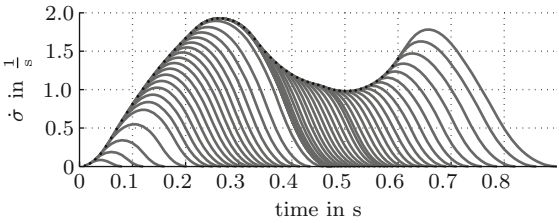



**Fig. 2.5** Path speed $\dot{\sigma} = x_2$ and all consecutively generated test trajectories over time

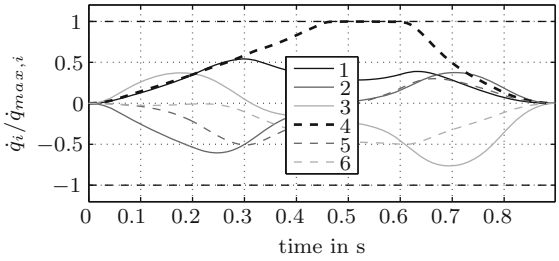



**Fig. 2.6** Normalized axis velocities

Corresponding axis speeds, torques and jerks are shown in Figs. 2.6, 2.7, and 2.8. Comparing the results reveals that always one of the limits is active (Sect. 2.3). Now, two state of the art offline methods are applied to the same problem to proof the performance of the presented algorithm. For the first method a B-Spline approximation of the function $\sigma(t)$ is used similar to Ref. [18]. For minimizing the equidistantly chosen time intervals of the knot vector, this results in a parametrization of the original optimal control problem (2.3), (2.4), and (2.5). The second method, a modified implementation of the multiple shooting method as described in Ref. [13], is also a direct method. Time optimization is achieved

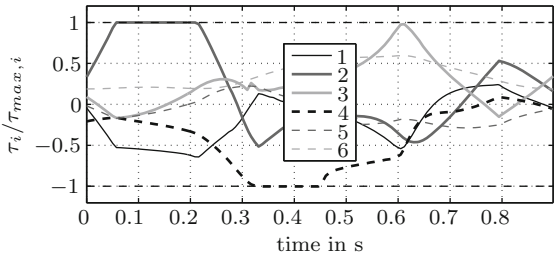



**Fig. 2.7** Normalized motor torques

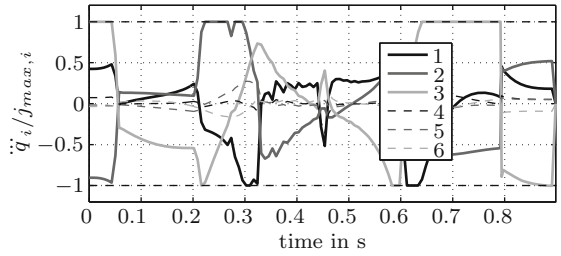



**Fig. 2.8** Normalized axis jerks

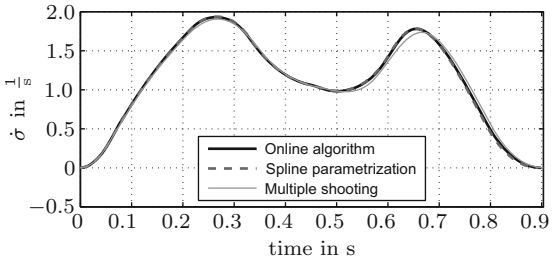



**Fig. 2.9** Comparison of path speed $\dot{\sigma} = x_2$ over time for three solution methods: (1) the presented algorithm, (2) a direct method using a spline parametrization and (3) a multiple shooting method

by minimizing the time interval between each multiple shooting node. Both methods are implemented in MATLAB and an SQP method is used to solve the nonlinear optimization problems.

The comparison of the three methods in Fig. 2.9 shows only small deviations which can be explained with different parameterizations and discretizations. A detailed comparison is given in Table 2.1. Although, due to MATLAB implementations, computation times have to be interpreted carefully, the comparison shows that the presented method seems to be very efficient and delivers satisfying results in realtime. Furthermore, the offline methods tend to fail for long geometric paths,

**Table 2.1** Comparison of optimization algorithms

| | Online alg. | Spline | Mult. shoot. |
|---|---|---|---|
| Final time | 0.8983 s | 0.8935 s | 9033 s |
| Num. points | 92 | 70 | 70 |
| CPU time | 0.49 s | 1.59 s | 34.3 s |
| $T_s$ | 0.01 s | 0.0129 s | 0.0127 s |

*Online alg.* is the presented algorithm, *Spline* is a method using a B-Spline parametrization and *Mult. shoot.* is a multiple shooting technique. *Final time* is the time $t_e$ to be minimized. *Num. points* are the number of discrete points (chosen for spline and multiple shooting). *Iterations* is the number of test trajectories for the presented algorithm and the number of SQP iterations for the other methods respectively. *CPU time* is the computation time for the whole path in MATLAB. The sampling time $T_s$ is chosen for the presented algorithm and a result for the other methods

whereas the presented algorithm is just running longer according to longer travel times.

## 2.6 Conclusion

The algorithm presented in this paper is an iterative online method for solving the problem of finding trajectories for the joint positions $\mathbf{q}(t)$ for a predefined geometric path. Therefore, time optimality and dynamic limits on joint speeds, accelerations, jerks and torques have to be considered. The method is directly derived from an optimality condition (Sect. 2.3) leading to lower and upper bounds for the path jerk (Sect. 2.3.2) as input for a discrete time system (Sect. 2.3.1). Further analysis on the existence of not allowed states show that a test for continuable states is necessary. Therefore, so called test trajectories are introduced, which form the basis of the presented algorithm (see Sect. 2.4.1). The main advantage of this method compared to other approaches is that a new path position (and therefore joint positions) can be outputted in each iteration step, which allows a realtime usage. An example shows that excellent results can be achieved, which is confirmed by a comparison with two other state of the art methods (Sect. 2.5).

For a future work deeper analysis on the optimality of the algorithm is planned.

**Acknowledgements** Support of the Austrian Center of Competence in Mechatronics (ACCM) is gratefully acknowledged.

## References

1. Bazaz SA, Tondu B (1999) Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators. Robot Autom Syst 29(4):257–268
2. Bellman RE, Dreyfus SE (1962) Applied dynamic programming. Princeton University Press, Princeton

3. Bobrow JE, Dubowsky S, Gibson JS (1985) Time-optimal control of robotic manipulators along specified paths. Int J Robot Res 4:3–17
4. Bremer H (2008) Elastic multibody dynamics: a direct Ritz approach. Springer-Verlag GmbH, Linz
5. Bryson AE, Ho YC (1975) Applied optimal control: optimization, estimation, and control. Hemisphere, Washington, DC
6. Chachuat B (2007) Nonlinear and dynamic optimization: from theory to practice. http://lawww.epfl.ch/page4234.html
7. Chand S, Doty K (1985) Online polynomial trajectories for robot manipulators. Int J Robot Res 4:38–48, Summer
8. Constantinescu D (2000) Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. J Robot Syst 17:233–249
9. Diehl M, Bock HG, Diedam H (2006) Fast direct multiple shooting algorithms for optimal robot control. Control 1:65–93
10. Franklin G, Powell JD, Workman ML (1998) Digital control of dynamic systems. Addison Wesley, Menlo Park
11. Hollerbach JM (1983) Dynamic scaling of manipulator trajectories. Am Control Conf 1983:752–756
12. Khalil W, Dombre E (20002) Modeling, identification & control of robots. Kogan Page Science, London
13. Leineweber DB, Bauer I, Bock HG, Schlöder JP (2003) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects. Comput Chem Eng 27(2):157–166
14. Macfarlane S, Croft E (2003) Jerk-bounded robot trajectory planning-design for real-time applications. IEEE Trans Robot Autom 19(1):42–52
15. Pardo-Castellote G, Cannon RH (1996) Proximate time-optimal algorithm for on-line path parameterization and modification. In: Proceedings of the IEEE international conference robotics automation, Minneapolis, USA, April 1996
16. Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko E (1962) The mathematical theory of optimal processes, International series of monographs in pure and applied mathematics. Interscience, New York
17. Shin K, McKay N (1985) Minimum-time control of robotic manipulators with geometric path constraints. IEEE Trans Autom Control 30(6):531–541
18. Verscheure D, Demeulenaere B, Swevers J, De Schutter J, Diehl M (2009) Time-optimal path tracking for robots: a convex optimization approach. IEEE Trans Autom Control 54 (10):2318–2327

# Chapter 3
# Constraint and Dynamic Analysis of Compliant Mechanisms with a Flexible Multibody Modelling Approach

**R.G.K.M. Aarts**

**Abstract** The models used in the conceptual phase of the mechatronic design should not be too complicated, yet they should capture the dominant system behaviour. Firstly, the awareness and possibly the avoidance of an overconstrained condition is important. Secondly, the models should reveal the system's natural frequencies and mode shapes in a relevant frequency range. For the control system synthesis the low frequent behaviour up to the cross-over frequency needs to be known. Furthermore, the closed-loop system can be unstable due to parasitic modes at somewhat higher frequencies.

In this chapter the applicability of a multibody modelling approach based on non-linear finite elements is demonstrated for the mechatronic design of a compliant six DOF manipulator. A kinematic analysis is applied to investigate the exact constrained design of the system. From dynamic models the natural frequencies and mode shapes are predicted and a state-space model is derived that describes the system's input-output relations. The models have been verified with experimental identification and closed-loop motion experiments. The predicted lowest natural frequencies and closed-loop performance agree sufficiently well with the experimental data.

## 3.1 Introduction

In high precision equipment the use of compliant mechanisms is favourable as elastic joints offer the advantages of no friction and no backlash. For the conceptual design of such mechanisms there is no need for very detailed and complex models that are time-consuming to analyse. Nevertheless the models should capture the

R.G.K.M. Aarts (✉)
Faculty of Engineering Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: R.G.K.M.Aarts@utwente.nl

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_3, 

dominant system behaviour which must include relevant three-dimensional motion and geometric non-linearities, in particular when the system undergoes large deflections. More specifically, we distinguish two phases in the modelling approach of which a kinematic design is the first phase. Typical design considerations for this phase aim at detecting and where necessary avoiding overconstrained or underconstrained design in line with so-called Exact Constraint Design principles [1–3]. The dynamic system performance is considered in the second design phase. It involves the computation of the natural frequencies and the accompanying mode shapes, which are closely related to the required closed-loop bandwidth and stability of the mechatronic system [4, 5].

In [6–9] we discussed the use of the SPACAR software for these design phases. It offers a multibody approach based on non-linear finite elements. The sound inclusion of the non-linear effects at the element level [10] appears to be very advantageous. Only a rather small number of elastic beam elements is needed to model e.g. wire flexures and leaf springs accurately. In particular for the kinematic analysis to check the constraints only a single flexible beam element is used for each flexure. In a dynamic analysis the natural frequencies are computed and more beam elements may be used to obtain more accurate results at higher frequencies or for larger deflections. The non-linear model can be linearised in a number of configurations throughout the complete operational range of the mechanism to obtain a series of locally linearised models in terms of the independent degrees of freedom, e.g. state space models for control system design [11]. Numerically efficient models are obtained as the number of independent degrees of freedom is rather small. Consequently, the approach is particularly well suited during the early (mechatronic) design phase, where time consuming computations would severely hamper the design progress.

This chapter is an extension of a paper earlier presented [8]. The modelling approach will be applied for the analysis and MIMO control system synthesis of a parallel kinematic precision manipulator with six kinematic degrees of freedom (DOF) as is described in the next section. Numerical results are presented in Sect. 3.3 and are verified with experimental data. Finally conclusions are drawn.

## 3.2 Six DOF Manipulator

Figure 3.1 shows a six DOF hexapod-like flexure-based manipulator [12]. It is an scaled-up version of a micromanipulator originally designed to be manufactured with MEMS technology. It has to translate and rotate the end effector in all directions. It is difficult to accurately measure the motion of the small micromanipulator which is not more than a few mm in size. Sensors can be integrated much easier in the scaled-up manipulator which has a largest outer dimension of 540 mm. The large version should give insight in the dynamic behaviour of the micromanipulator and therefore the restrictions in the mechanical design resulting from the MEMS fabrication method have been preserved.

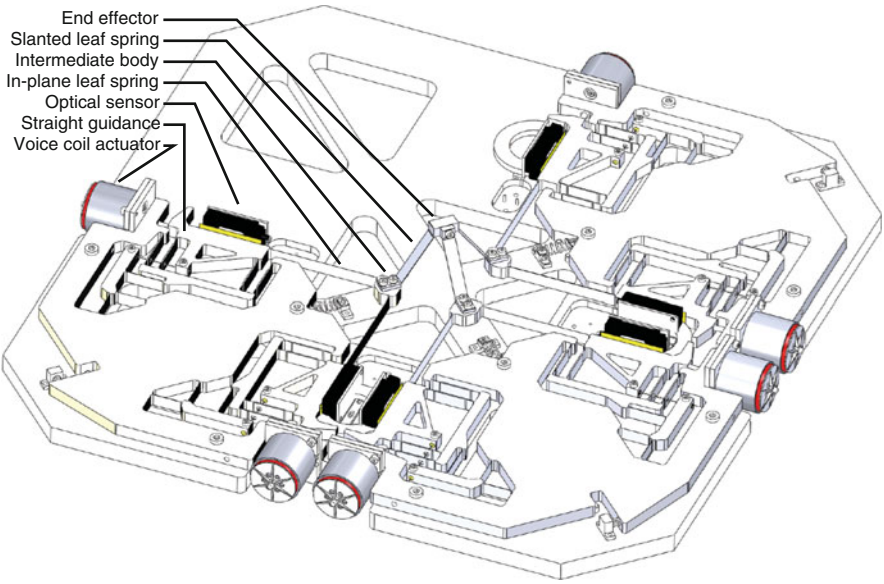



**Fig. 3.1** Six DOF hexapod-like manipulator with flexible joints [12]

In the scaled-up manipulator six voice coil actuators (VCMs) are applied to drive the position and orientation of the end effector. In the MEMS design it is essential that each actuator is constrained to a purely translational motion. In the scaled-up version this motion is also enforced by straight guidances that assure that the motion of each VCM is exactly in one in-plane direction. Each VCM is equipped with a contact-free optical incremental encoder to measure the actuator displacement for colocated feedback control. The motions of a pair of VCMs are transferred via in-plane leaf springs to an intermediate body, such that this body can move in the in-plane directions. In total three of these intermediate bodies support three slanted leaf springs that are connected to the end effector. In this way the three times two in-plane actuated translations of the intermediate bodies enable translations and rotations of the end effector in all six DOF. E.g. the horizontal translations of the end effector are realised with identical motions of all three intermediate bodies. To accomplish a vertical translation of the end effector, the three intermediate bodies move radially towards the centre of the set-up. These motions and the rotations are outlined in more detail by Brouwer et al. [12]. In general, the relations between the linear VCM displacements and the position and orientation of the end effector are highly non-linear. These relations can be measured with a sensor system that is mounted on the end effector. This sensor system (not shown in Fig. 3.1) includes an optical sensor to measure the displacement in one long-stroke direction, while the parasitic displacements in the perpendicular directions and the rotations are measured with capacitive sensors.

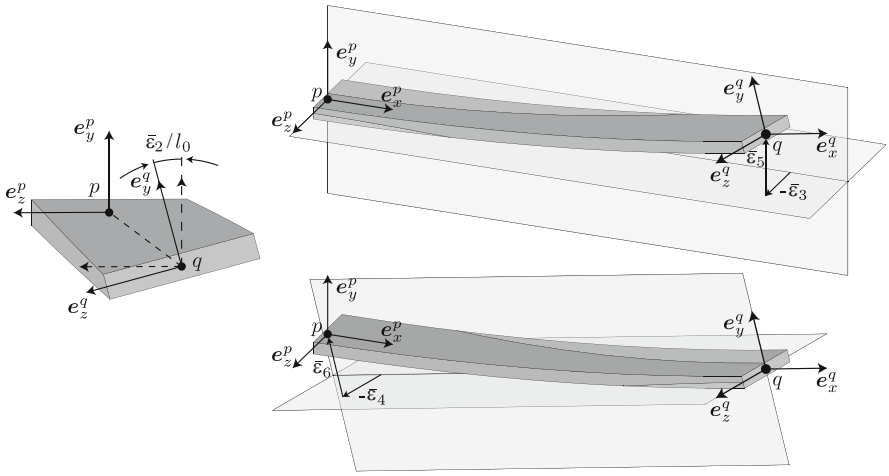



**Fig. 3.2** Deformations $\varepsilon_2$–$\varepsilon_6$ of the spatial beam element (Reprinted from [14])

## 3.3 Numerical Modelling

A numerical model of the manipulator needs to account for the flexures in the system. More specifically each straight guidance consists of leaf springs and a wire spring. The end effector is mounted on the three slanted leaf springs. In the modelling approach implemented in SPACAR flexible beam elements are used for all flexures. This beam element will be outlined first before the kinematic and dynamic analyses are presented.

### *3.3.1 Spatial Flexible Beam Element*

The location of the beam element is described by the positions of the end nodes $p$ and $q$, as well as their orientations. Essential is the definition of physically meaningful deformation modes of the element that are invariant for rigid body motions of the element. As there are 12 independent nodal coordinates and six rigid body degrees of freedom, six independent deformation modes can be defined. For the spatial flexible beam one deformation mode coordinate $\varepsilon_1$ is taken to describe the elongation, $\varepsilon_2$ for torsion and four modes $\varepsilon_{3\text{–}6}$ for the bending deformations of the element [10, 13]. Figure 3.2 illustrates five of these deformation modes. The deformation mode coordinates are defined in such a way that geometrically non-linear effects due to interaction between deformation modes are included. Consequently, accurate models can be obtained with a relatively small numbers of elements even for the case when large deflections are considered [10, 13]. Each of the deformation mode coordinates can be defined to be constrained or released.

If a deformation mode coordinate is released, i.e. not constrained, constitutive equations have to be specified for the stress resultants, which are dual to the deformations. These constitutive equations may express simply linear elastic behaviour based on the element stiffness properties.

### 3.3.2 Kinematic Model and Exact Constrained Design

Numerical models of the system can be made with a varying level of complexity. With a *kinematic* SPACAR model it can be verified that the manipulator satisfies exact constraint design. In this model each wire flexure and leaf spring is modelled with a single flexible beam element. All deformation modes with a high stiffness are considered to be rigid, i.e. having constrained deformation mode coordinates. The deformation modes with low stiffnesses are allowed to deform. Then it appears that a Jacobian matrix can be assembled which must be square and full rank in order to satisfy exact constraint design: otherwise the system is underconstrained or overconstrained [7, 9].

The straight guidances of the manipulator, Fig. 3.3, are overconstrained by design to increase the stiffness in the out-of-plane direction. This is confirmed in the kinematic analysis and these parts are manufactured accurately to minimise the internal stresses [12]. A six DOF kinematic model confirms the exact constraint design of the end effector motion, Fig. 3.4.

Note that for this kinematic analysis the masses and stiffnesses do not play a role. These are of course relevant in the dynamic analysis to be discussed next.

### 3.3.3 Dynamic Model and Natural Frequencies with Mode Shapes

Natural frequencies and mode shapes are obtained from *dynamic* models. The simplest dynamic model is derived from the kinematic model outlined above in which mass and stiffness properties are added. In the applied modelling approach the non-linear equations of motion can be linearised in any valid configuration of the system. From the mass and stiffness matrices the (configuration dependent) natural frequencies and mode shapes are computed. A state space model is derived after defining the system's inputs, the VCM forces, and outputs, the colocated sensor positions. As the simplest dynamic model has six DOF, only the six lowest natural frequencies of the manipulator can be obtained from this model and a twelfth order state space model is found.

For control system synthesis also higher natural frequencies and their mode shapes must be known [4]. These so-called parasitic modes involve deformations in the directions of the larger stiffnesses. In the dynamic model they can be accounted for by releasing deformation mode coordinates associated with deformations in

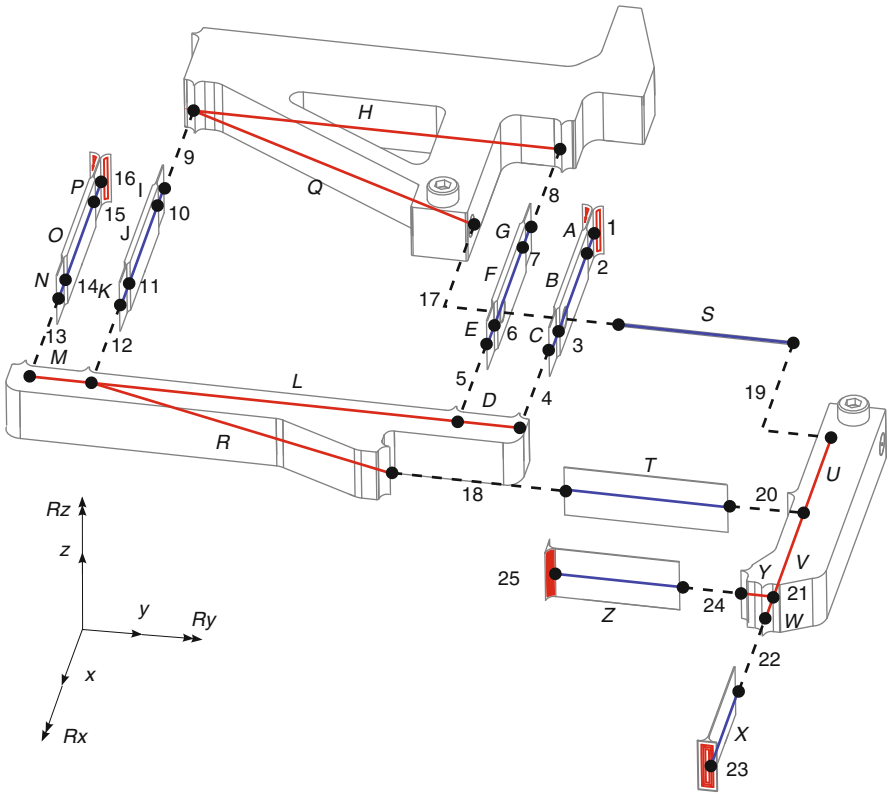



**Fig. 3.3** Schematic model of each straight guidance. The lines in the solid parts represent rigid elements. The lines in the flexures ABC, EFG, IJK, NOP, S, T, X and Z are flexible beam elements. The dashed lines are connections between elements that are apart for a clearer view. In the points *1*, *16*, *23* and *25* the guidance is fixed to the world. The motion of body H is guided. Lever U assures that the stroke of the intermediate body L is half of the stroke of body H (From [15])

these directions. In the previous six DOF model these deformations were prescribed zero and now these released deformation mode coordinates give rise to additional degrees of freedom. Furthermore, the system should be evaluated in configurations throughout the manipulator's workspace. The six deformation modes of the flexible beam element offer only an accurate approximation for a limited set of element deformations. If more complex deformations are expected, the approximation can be improved by increasing the number of elements in each flexure. Obviously, both improvements of the dynamic model result in an increased number of DOF.

For the considered manipulator a model has been made in which three or four beam elements are used for each wire flexure of leaf spring. This model has 870 DOF which result in many natural frequencies that are far outside the frequency range of interest. To reduce the number of DOF the model is first simplified by reducing the number of beam elements that is used for the flexures. If the lower natural frequencies of the reduced order model are identical or close to the natural

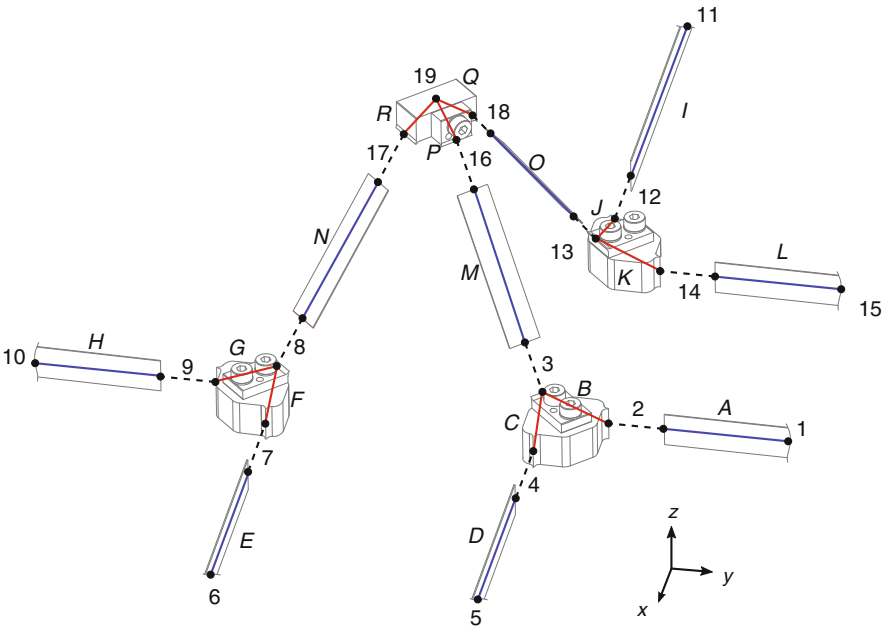



**Fig. 3.4** Schematic model of the end effector. The lines in the solid intermediate bodies and end effector represent rigid elements. The lines in the flexures A, D, E, H, I, L, M, N and O are flexible beam elements. The dashed lines are connections between elements that are apart for a clearer view. In the points *1*, *5*, *6*, *10*, *11* and *15* purely translational motions are prescribed which cause in-plane motion of the intermediate bodies *C*, *G* and *K*. As a result, the out-of-plane leaf springs *M*, *N* and *O* move the end effector (points *16–19*) in all translational and rotational degrees of freedom (From [15])

frequencies of the 870-DOF model in this range, the simplification is accepted. In this way the number of DOF could be reduced to 420. A further simplification is possible by constraining deformations. The longitudinal stiffness of the flexures is rather high and it appears that a model with all elongations $\varepsilon_1$ prescribed zero results in 315 DOF without loss of accuracy. Similarly also part of the bending deformation modes with a high stiffness can be considered rigid and finally a 237-DOF model is obtained. Table 3.1 lists the numerical values of the ten lowest natural frequencies of both the extended 870-DOF and the reduced 237-DOF models. As can be seen in the table the lowest six natural frequencies of the reduced model are almost identical to the natural frequencies of the large model. For the higher natural frequencies somewhat larger differences are found. For the control system synthesis, in particular the seventh natural frequency is relevant which differs by about 6%. In Fig. 3.5 these natural frequencies can be recognised as the peaks in the graph of the system's singular values or principal gains as functions of the frequency. In this analysis the VCM forces are the system's inputs and the colocated sensors are the outputs. The lowest natural frequencies are damped due to the actuator's back-EMF.

**Table 3.1** First ten natural frequencies (in rad/s) of the large and reduced models as well as the experimentally identified natural frequencies. Note that the experimentally observed 7th and 8th natural frequency are not present in the models

| Mode | 870-DOF | 237-DOF | Mode | Exp. |
|---|---|---|---|---|
| 1 | 59.3 | 59.3 | 1 | 55.3 |
| 2 | 59.9 | 59.9 | 2 | 56.0 |
| 3 | 84.3 | 84.5 | 3 | 79.6 |
| 4 | 86.8 | 86.8 | 4 | 83.0 |
| 5 | 122.0 | 122.2 | 5 | 116.1 |
| 6 | 124.6 | 125.1 | 6 | 120.2 |
| | | | 7 | 225 |
| | | | 8 | 285 |
| 7 | 658 | 697 | 9 | 565 |
| 8 | 844 | 917 | | |
| 9 | 908 | 1001 | 10 | 1050 |
| 10 | 1074 | 1303 | | |

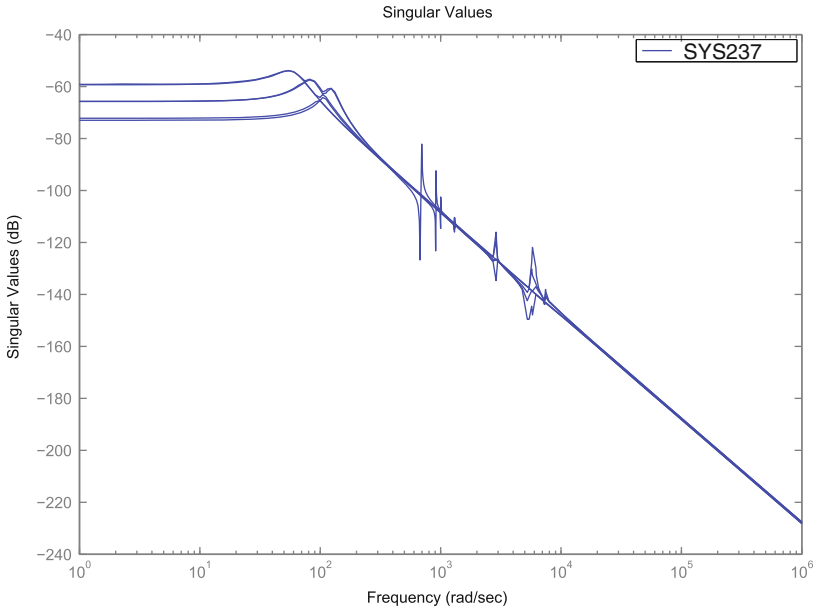



**Fig. 3.5** Singular values of the transfer matrix of the 237-DOF SPACAR-model near the equilibrium configuration (From [15])

The linearised models of the mechanical system are well-suited for control system synthesis. The following steps are taken. At first the cross-over frequency of the feedback controller is determined from performance requirements. Assuming this cross-over frequency will be well below the unwanted higher natural frequencies, the closed-loop performance can be evaluated from the controller combined with the low frequent behaviour of the mechanical system [4], i.e. the six lowest natural frequencies. For this purpose a linearised six DOF model that accounts for the lowest six natural frequencies in Table 3.1 is well-suited. As an example we consider a PID-like feedback controller that should track a third order motion profile during 1 s with an error of less than 0.1% of the amplitude. This can be accomplished with a cross-over frequency of about 300 rad/s. Secondly, the closed-loop performance can be improved with feedforward control. A feedforward control input can be computed by applying a stable inverse approximation of a low frequent model of the mechanical system to the desired motion profile.

Finally the robust stability of this closed-loop system can be evaluated. In particular the first parasitic natural frequency may violate stability requirements in an $\mathcal{H}_\infty$ controller design strategy [4]. Obviously for this purpose a model of the mechanical system like the 237-DOF model is needed that is sufficiently accurate above the cross-over frequency. This model can also be used in closed-loop simulations to validate the controller design.

## 3.4 Experimental Results

An experimental set-up with the manipulator of Fig. 3.1 has been realised. As outlined in Sect. 3.2 it is actuated with six VCMs. Colocated sensors measure the actuator displacements. MIMO system identification has been carried out with a black-box multivariable output error subspace (MOESP) model identification method [15–17]. A 21st order model is found that identifies the lowest natural frequencies as well as the first parasitic modes. These natural frequencies are included in Table 3.1 and are combined with the 237-DOF model in Fig. 3.6. It appears that the six lowest natural frequencies agree quite well between the numerical model and experimental data. Also the natural frequency of the first parasitic mode agrees reasonably well. However, two additional natural frequencies are found in the identification that are not included in the models. Probably these modes arise from suspension modes of the frame that are not accounted for in the numerical models. In Fig. 3.6 these modes are visible, but their amplitudes are rather small. Overall it is concluded that the numerical models provide an adequate prediction of the experimental results.

The designed feedback and feedforward controller has been tested for a motion of the end effector of 6 mm displacements in the horizontal $x$, $y$-plane. Figure 3.7 shows the tracking error of the actuator displacements during this motion. It appears

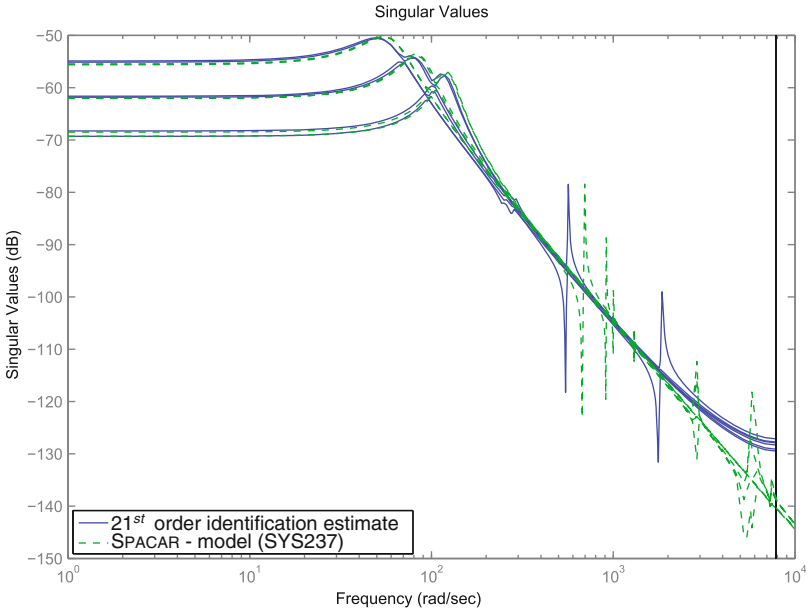



**Fig. 3.6** Singular values of the transfer matrix of the 21st order identification estimate and the SPACAR-model near the equilibrium configuration (From [15])

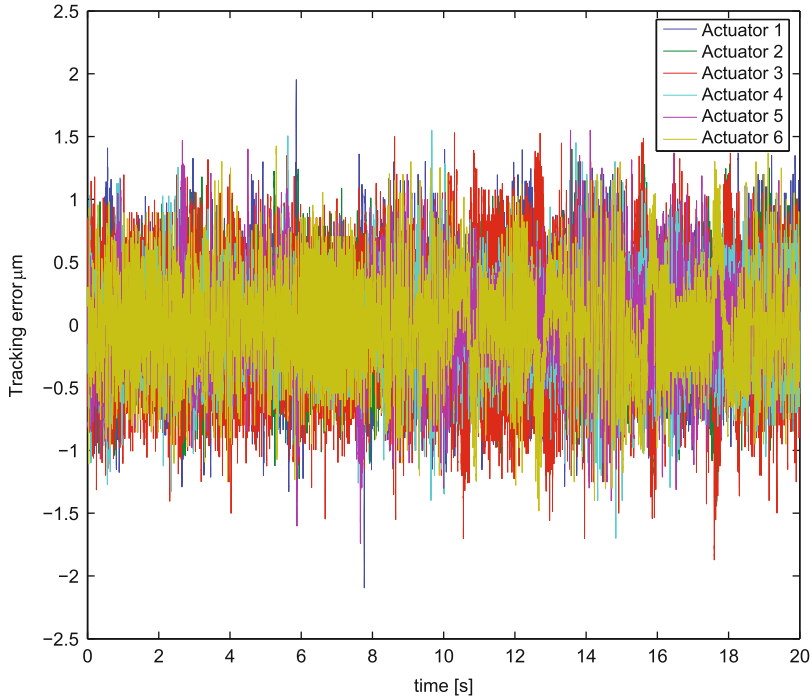



**Fig. 3.7** Measured tracking error of the actuator displacements during a 6 mm displacement of the end effector in the $x$, $y$ -plane (From [15])

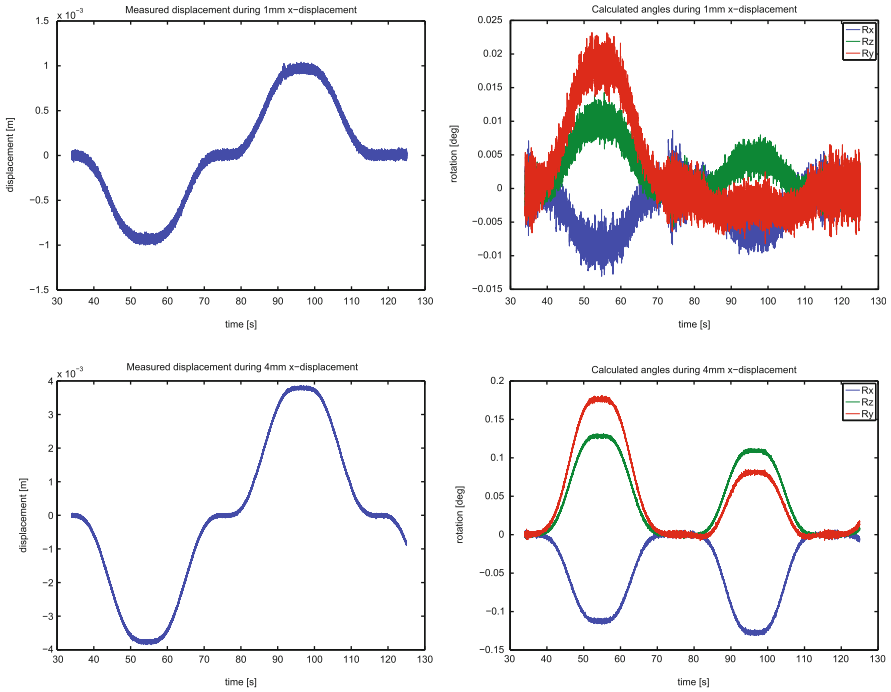



**Fig. 3.8** Measurements of the end effector motion during 1 mm (*top*) and 4 mm (*bottom*) $x$-displacements of the end effector. The left graphs show the long stroke motion in the $x$ direction; the right graphs show all three rotations of the end effector for both displacements (From [15])

that they remain below the desired 0.1% although the signal is quite noisy. This is to a large extend caused by a 50 Hz disturbance from the mains.

Finally, the motion of the end effector has been analysed with the sensor mounted on the end effector. This sensor can measure a long stroke in one direction and small deviations in the other directions as well as rotations. The $x$ axis of the coordinate system is aligned with the direction of the long stroke. The linearised manipulator model has been used to compute the actuator displacements needed for linear displacements of the end effector in the $x$ direction of 1 mm and 4 mm, respectively. Figure 3.8 shows the actually measured motion of end effector. It is found that the real displacement matches reasonably with the intended motion, but it is somewhat smaller than expected. Furthermore, unwanted rotations are observed. To some extend both effects can be caused by a small misalignment between the coordinate systems of the manipulator and the sensor. However, it is also noted that the deviations increase more than linearly when the amplitude of the end effector displacement is increased. This could be caused by the non-linear behaviour of the manipulator which is not yet included in the model currently used to compute the needed actuator displacements.

## 3.5 Conclusions

The design of a mechatronic system of the six DOF compliant manipulator in Fig. 3.1 demonstrates the proposed modelling approach for this purpose. The formulation is based on a nonlinear finite element description for flexible multibody systems. The flexible beam elements account for geometric nonlinear effects such as geometric stiffening and interaction between deformation modes. Flexible joints like wire flexures and leaf springs can be modelled adequately using only a few number of flexible beam elements. In this way, a rather low dimensional system description can be obtained which includes the non-linear behaviour that occurs at large deflections.

In a kinematic analysis only a single flexible beam element is used for each wire and sheet flexure and the exact constrained design of the system is examined. In particular overconstrained conditions are detected and if necessary the design can be modified to avoid these overconstraints. For the dynamic analysis a maximum of four flexible beam elements is used for each flexure. The number of DOF is reduced by prescribing deformations with high stiffness and in rigid parts to be zero. In any configuration of the manipulator the natural frequencies and mode shapes can be computed. Furthermore, an input-output state space model can be derived to design and evaluate the control system. The modelling approach is well suited for mechatronic design, i.e. the mechanical design as well as control system synthesis.

**Acknowledgements** The author acknowledges the contributions from Steven Boer, Dannis Brouwer, Johannes van Dijk, Ben Jonker and Jaap Meijaard to the design, modelling and control as outlined in this chapter. Martijn Huijts and Sytze Spijksma are acknowledged for the design, analysis and testing of the manipulator.

## References

1. Blanding DL (1999) Exact constraint: machine design using kinematic principles. ASME Press, New York
2. Hale LC (1999) Principles and techniques for designing precision machines. Ph.D. thesis, University of California, Livermore
3. Soemers HMJR (2010) Design principles for precision mechanisms. T-point print, Enschede
4. van Dijk J, Aarts RGKM, Jonker JB (2012) Analytical one parameter method for PID motion controller settings. In: IFAC conference on advances in PID control – PID'12, Brescia, 28–30 Mar 2012
5. Xu JX (2008) New lead compensator designs for control education and engineering. In: Proceedings of 27th Chinese control conference, 16–18 July 2008
6. Aarts RGKM, van Dijk J, Jonker JB (2009). Efficient analyses for the mechatronic design of mechanisms with flexible joints undergoing large deformations. In: ECCOMAS thematic conference multibody dynamics 2009, Warsaw University of Technology, Warsaw, June 29–July 2

7. Aarts RGKM, van Dijk J, Jonker JB (2010) Flexible multibody modelling for the mechatronic design of compliant mechanisms. In: The 1st joint international conference on multibody system dynamics, Lappeenranta, 25–27 May 2010
8. Aarts RGKM, van Dijk J, Brouwer DM, Jonker JB (2011) Application of flexible multibody modelling for control synthesis in mechatronics. In: Multibody dynamics 2011, ECCOMAS thematic conference, Brussels, 4–7 July 2011
9. Aarts RGKM, Boer SE, Meijaard JP, Brouwer DM, Jonker JB (2011) Analyzing overconstrained design of compliant mechanisms. In: Proceedings of ASME 2011 international design engineering technical conference & computers and information in engineering conference – IDETC/CIE 2011, Washington, DC, 28–31 Aug 2011
10. Jonker JB, Meijaard JP (2009) Definition of deformation parameters for the beam element and their use in flexible multibody system analysis. In: ECCOMAS thematic conference multibody dynamics 2009, Warsaw University of Technology, Warsaw, June 29–July 2
11. Jonker JB, Aarts RGKM, van Dijk J (2009) A linearized input-output representation of flexible multibody systems for control synthesis. Multibody Syst Dyn 21(2):99–122
12. Brouwer DM, de Jong BR, Soemers HMJR (2010) Design and modeling of a six DOF's MEMS-based precision manipulator. Precis Eng 34(2):307–319
13. Meijaard JP, Brouwer DM, Jonker JB (2010) Analytical and experimental investigation of a parallel leaf spring guidance. Multibody Syst Dyn 23(1):77–97
14. Boer SE, Aarts RGKM, Brouwer DM, Jonker JB (2010) Multibody modelling and optimization of a curved hinge flexure. In: The 1st joint international conference on multibody system dynamics, Lappeenranta, 25–27 May 2010
15. Spijksma SA (2010) Analysis and MIMO control of a 6 DOFs parallel kinematic precision manipulator. M.Sc. thesis, Report number Wa-1293, University of Twente, Enschede
16. Nijsse G (2006) A subspace based approach to the design, implementation and validation of algorithms for active vibration isolation control. PhD thesis, University of Twente, Enschede
17. van Overschee P, de Moor B (1994) N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30(1):75–93

# Chapter 4
# Sensor Data Fusion for the Localization and Position Control of One Kind of Omnidirectional Mobile Robots

**Peter Eberhard and Qirong Tang**

**Abstract** This contribution deals with the problem of sensors and sensor data fusion for mobile robots localization and position control. For this, the robot internal odometry is first corrected and used for position control. Then, an extended Kalman filter based on the corrected odometry and a novel North Star navigation system is designed in a distributed manner. The estimated information from the extended Kalman filter is feed back to the desired poses for further accelerating and precising the position control process. Finally, after the analysis of data flows and uncertainties, the whole developed scheme is verified by experiments on an omnidirectional mobile robot.

## 4.1 Introduction

Localization is a fundamental and key issue for mobile robots since it is the prerequisite for many abilities, e.g., path planning, navigation and execution of tasks. Besides traditional sensor based methods, there are also many new theoretical approaches for solving this problem such as, using probabilistic [1, 2], topology [3, 4, 5] and fuzzy logic [6, 7]. No matter what kind of methods are used, when facing real robots one must deal with the used sensors and the resulting sensors data. Although some methods are very mature in theory, it is still difficult to utilize and validate them under realistic conditions or with combinations of other strategies. Sources of difficulties come, e.g., from restrictions like sensors precision, environment noise, uncertainties of the robot system. For these reasons, researchers rarely use just one method or sensor for robot localization, but instead hybrid strategies which are based on several kinds of methods based on hardware and software.

P. Eberhard (✉) • Q. Tang
Institute of Engineering and Computational Mechanics, Pfaffenwaldring 9,
Stuttgart 70569, Germany
e-mail: peter.eberhard@itm.uni-stuttgart.de; qirong.tang@itm.uni-stuttgart.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_4, 

Usually the localization is mobile robots oriented, typically for differentially driven mobile robots. However, in recent years omnidirectional mobile robots attracted researchers interest due to their unique features. This study presents methods for the localization of one kind of omnidirectional mobile robots, the Festo Robotino, and the scheme is valid for all of the individuals in the robot group which is used for our purpose of searching a target in an environment. Some researches focus on similar robots and localization questions, e.g., [8, 9], but they pay more attention to the path planning issue. The investigations described in [10, 11, 12, 13, 14] focus on multi-robot collaborative localization but unfortunately they neither consider to improve the accuracy of sensors nor individual measurements before used for localization. Thus, they have to pay more efforts on data post-processing. The robot trajectories in these studies are neither optimal nor under an unified movement guidance. To the contrary, they directly use the available sensors even with high noises or uncertainties. The study [14] contributes its decentralized architecture and [10] uses a minimum entropy approach to minimize sensor uncertainty. The work in [13] is oriented for an outdoor environment where the global positioning system (GPS) is applied. However, GPS doesn't effectively work inside of the buildings. The research in [11] and [12] performs localization by relying on the probabilistic approaches of Markov localization and maximum likelihood estimation, respectively. As a result, these studies for multi-robot localization greatly increase the burden for positioning and also involve more interferences. Other researches study differentially driven robots and there are only few researches that concern the localization for omnidirectional mobile robots with consideration of robot sensors and swarm behavior first.

Section 4.2 introduces some common concepts and states the reason of localization, then the overall position control scheme and the used sensors are described in Sect. 4.3. After the introduction of Kalman filters the detailed sensor data fusion and localization processes are shown in Sect. 4.4. Experiments with a real robot and results analysis are performed in Sect. 4.5 while Sect. 4.6 gives conclusions.

## 4.2 Some Common Concepts

For a better understanding, it is necessary to introduce some common concepts such as, e.g., control points and measurements, localization and navigation, controller and robots position control. A classical feedback control diagram is illustrated in Fig. 4.1 which is used in the following.

### *4.2.1 Control Points and Measurement*

The control points in our case are referred to the desired position points which are generated by robot swarm under mechanical PSO algorithm. The controller manipulate points sometimes are also called control variables. They are generated by the

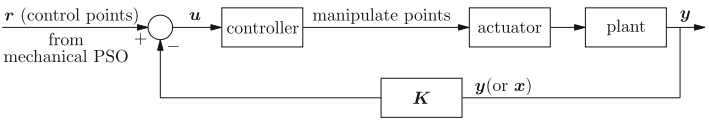



**Fig. 4.1** Classical feedback control

system controller and act on the controlled plant directly or through an actuator to manipulate the concerned variable to approach the reference variable.

The controlled plant show certain behavior, and the system should have some sensors to measure it. Thus, the system needs some devices to perform measurements, then compare the obtained information to the references. In this study, the sensors are the robots internal odometer and the external North Star system.

### *4.2.2 Navigation Strategies and Classification*

For mobile robot research one can not avoid the topic of localization and navigation. The former one answers the question 'Where am I?' to the robot, i.e., provides the position, orientation and additional information like the environment and info about other robots. The latter one is a comprehensive strategy which guides the robot (hopefully in an optimal way) and includes also environment detection and obstacle avoidance yielding a collision free path. Table 4.1 lists some basic methods for localization and navigation. Details can be found in the mentioned publications or in [15].

Robot motion planning belongs to robot navigation which includes location, path planning, and obstacle avoidance. However, these three different topics are often considered at the same time and are summarized in the term 'motion planning' in practical applications.

### *4.2.3 Controller and Robot Position Control*

According to the measurements by sensors and the reference values, see Fig. 4.1, the controller adjusts values of the actuator so as to alter the status of the controlled plant.

Robot position control is based on the fact that a controller is considered which focuses on robot positioning, trajectory tracking and so on. Please distinguish this to the general position control concept which usually takes into account a servo position control with emphasize on controlling the pulse inputs.

### *4.2.4 Why Do We Need Localization?*

One of the project goals is to use a group of omnidirectional mobile robots to search a target in an environment and the main guidance mechanism is based on the

**Table 4.1** Robot navigation strategies and motion planning

| | | | |
|---|---|---|---|
| localization | relative | dead reckoning | |
| | absolute | imaging, laser, GPS | |
| path planning | local | APF, genetic algorithms, fuzzy logic | |
| | global | environment modeling | graph methods, free-space methods, grid methods |
| | | path search | A* algorithms [23], D* optimal algorithms [24] |
| obstacle avoidance | VFH [25], APF [26], VFH+ [27], VFH* [28] | | |

mechanical Particle Swarm Optimization (PSO) which is an extension from the basic PSO and includes some of robots mechanical properties [16]. This scheme uses the mechanical PSO to generate the search trajectory. Then the robots further perform fine tuning for obstacle avoidance and mutual avoidance locally. A main demand of this method is, that it requires the robots current velocities, the self-best positions as well as the swarm-best positions, i.e., it heavily relies on the pose of each robot. Thus, the localization with acceptable accuracy becomes important.

## 4.3 Position Measurement and Control

To successfully perform some motions, e.g., the trajectory tracking on mobile robots where the trajectory is generated by a mechanical PSO algorithm, one must also consider the robots localization abilities. Unfortunately, in reality a controller can not arbitrarily accurate command the robots to the positions and usually the robots lack accurate positioning functionality. In some cases it is even very difficult to obtain a precise position from robots.

Originally Robotinos can only roughly obtain their position information by using motor encoders if used without help from external positioning equipment. Such a measurement, however, can not get meet the requirement. Therefore, this study builds a systematic hybrid strategy for the localization. Combined with control techniques, it enables to drive robot to a desired position within an acceptable accuracy. This section first of all shows the overall scheme for robot position control and then goes on with a detail research of the sensors involved in the measurements for localization.

### *4.3.1 Overall Position Control Scheme*

The overall scheme for the localization and position control of a single Robotino is shown in Fig. 4.2.

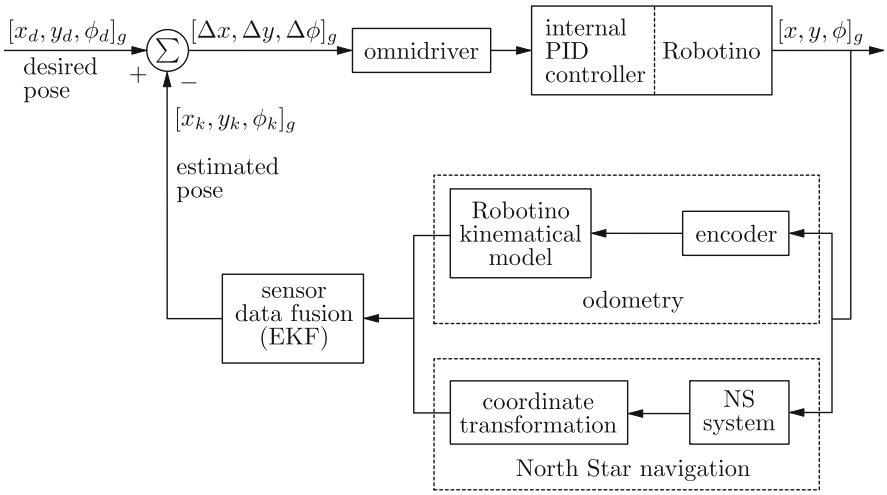



**Fig. 4.2** Scheme of accurate localization

The scheme consists of one internal controller and one external controller. First of all, a partially fixed PID controller which can directly control the DC motor is adjusted and used as an internal controller. The robot's 'actual' position and orientation are measured by not only on-body encoders but also an external North Star system. The measured pose is then processed in an extended Kalman filter (EKF). Data fusion in the EKF gives an estimated position and orientation which are compared to the desired. This forms the closed-loop feedback control which contains here also the external controller. The resulting deviation $[\Delta x\ \Delta y\ \Delta\phi]_g$ goes on to drive the robot. By using such a hybrid strategy the localization accuracy of the robots can be improved. The scheme is valid for all the robots in the group since each robot runs on its own.

### 4.3.2 State Variable Feedback

From Fig. 4.1 one can see that a proportional feedback control law

$$\boldsymbol{u} = -\boldsymbol{K} \cdot \boldsymbol{y} + \boldsymbol{r} \tag{4.1}$$

is used. This scheme is also known as output feedback. However, the output feedback design has many difficulties for pole placement, see details, e.g., in [17] and [18]. Another basic control scheme is to use all the states as outputs, then it becomes state variable feedback with the control law

$$\boldsymbol{u} = -\boldsymbol{K} \cdot \boldsymbol{x} + \boldsymbol{r}. \tag{4.2}$$

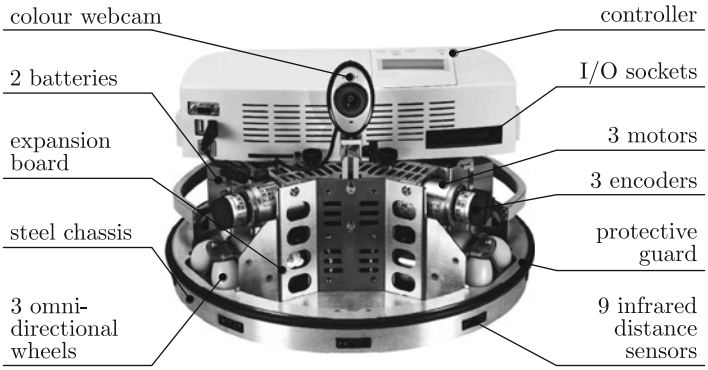



**Fig. 4.3** Festo Robotino and its main components

Benefiting from high quality sensors and techniques like Kalman filters and observers, to measure all of the state variables directly or indirectly becomes possible. Most important, the state variable feedback is simple for pole placement, see also in [17] and [18]. In this study, it considers the robots pose (2D position, one orientation) as state variables. Actually, by a reasonable observer, it also can include pose rates. In the following this study will focus on the sensors which are used for state measuring.

### *4.3.3 Internal Odometer and Its Enhancement*

Odometry is the measurement of wheel rotation with use of many different methods that are integrated in the drive system and continually updates with incremental wheel information. The position and orientation then can be determined easily by time integration added to the previously known position.

#### 4.3.3.1 Odometry Mechanism

The Festo Robotino is a holonomic mobile robot which contains three omni drive units with 120° between each. The robot and its main components are shown in Fig. 4.3, the structure of Robotino base and coordinates can be seen in Fig. 4.4.

After simple derivation one can get the kinematic relation between wheel speed and global velocity by

$$\begin{bmatrix} \dot{x}_g(t) \\ \dot{y}_g(t) \\ \dot{\phi}_g(t) \end{bmatrix} = r \begin{bmatrix} -\sin(\phi_g(t)+\alpha_1) & \cos(\phi_g(t)+\alpha_1) & R \\ -\sin(\phi_g(t)+\alpha_2) & \cos(\phi_g(t)+\alpha_2) & R \\ -\sin(\phi_g(t)+\alpha_3) & \cos(\phi_g(t)+\alpha_3) & R \end{bmatrix}^{-1} \cdot \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{bmatrix} \quad (4.3)$$

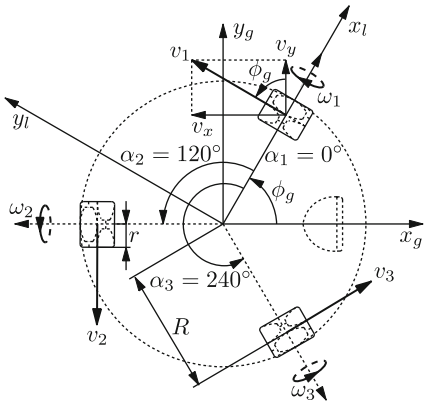



**Fig. 4.4** Geometry diagram and kinematic relations of Robotino base

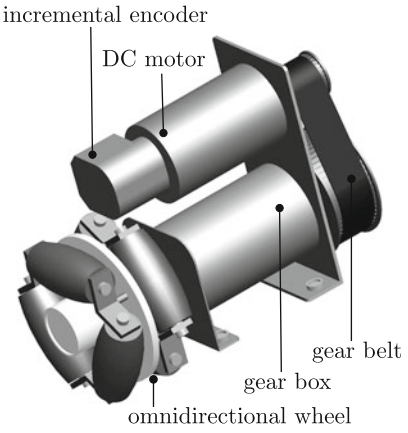



**Fig. 4.5** Robotino drivetrain

where $r$ is the wheel's radius, $R$ is the distance from the wheel's center to the center of the robot, and $\omega_i(t)(i = 1, 2, 3)$ are the wheels angular velocities. The geometry angle $\phi_g$ and $\alpha_i(i = 1, 2, 3)$ are illustrated in Fig. 4.4, and $[\dot{x}_g \ \dot{y}_g \ \dot{\phi}_g]^T$ represents the global velocity of the robot. Odometry can estimate the robots position and orientation over time by

$$\begin{bmatrix} x_g(t_1) \\ y_g(t_1) \\ \phi_g(t_1) \end{bmatrix} = \begin{bmatrix} x_g(t_0) \\ y_g(t_0) \\ \phi_g(t_0) \end{bmatrix} + \int_{t=t_0}^{t_1} \begin{bmatrix} \dot{x}_g(t_0) \\ \dot{y}_g(t_0) \\ \dot{\phi}_g(t_0) \end{bmatrix} dt. \tag{4.4}$$

The so called 'odometry' in this study is based on the robots motor encoders (see the drivetrain of the Robotino in Fig. 4.5) and the kinematics relationships (4.3) and (4.4). The rotation of robot wheels is measured with the highest possible resolution.

In each small time step the distance driven by a robot is calculated by (4.4). Thus, the current position of the controlled robot relative to its starting point can be calculated. The time integration is actually calculated as counting the number of beam interruptions caused from the toothed interrupter disc. So, if one considers this in a discrete way together with first order Taylor series expansion, it results in

$$\begin{bmatrix} x_g(k) \\ y_g(k) \\ \phi_g(k) \end{bmatrix} = \begin{bmatrix} x_g(k-1) \\ y_g(k-1) \\ \phi_g(k-1) \end{bmatrix} + \Delta t\, r\boldsymbol{T}(k-1) \cdot \begin{bmatrix} \omega_1(k-1) \\ \omega_2(k-1) \\ \omega_3(k-1) \end{bmatrix} \tag{4.5}$$

where $\Delta t$ is the size of time step, typically in our case $\Delta t = 0.01\text{s}$. The matrix $\boldsymbol{T}(k-1)$ includes the structure matrix and the rotation matrix from robot local coordinates to global coordinates which can be governed by

$$\begin{aligned} \boldsymbol{T}(k-1) &= \begin{bmatrix} -\sin(\phi_g(k-1)+\alpha_1) & \cos(\phi_g(k-1)+\alpha_1) & R \\ -\sin(\phi_g(k-1)+\alpha_2) & \cos(\phi_g(k-1)+\alpha_2) & R \\ -\sin(\phi_g(k-1)+\alpha_3) & \cos(\phi_g(k-1)+\alpha_3) & R \end{bmatrix}^{-1} \\ &= \left( \begin{bmatrix} -\sin(\alpha_1) & \cos(\alpha_1) & R \\ -\sin(\alpha_2) & \cos(\alpha_2) & R \\ -\sin(\alpha_3) & \cos(\alpha_3) & R \end{bmatrix} \right. \\ &\quad \left. \cdot \begin{bmatrix} \cos(\phi_g(k-1)) & \sin(\phi_g(k-1)) & 0 \\ -\sin(\phi_g(k-1)) & \cos(\phi_g(k-1)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} \cos(\phi_g(k-1)) & -\sin(\phi_g(k-1)) & 0 \\ \sin(\phi_g(k-1)) & \cos(\phi_g(k-1)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\sin(\alpha_1) & \cos(\alpha_1) & R \\ -\sin(\alpha_2) & \cos(\alpha_2) & R \\ -\sin(\alpha_3) & \cos(\alpha_3) & R \end{bmatrix}^{-1}. \end{aligned}$$

#### 4.3.3.2 Odometry Error Correction

Robot odometry is used for pose measurement since it is an internal 'sensor' system and is convenient for practical utilization. However, this method only yields good performance for short distance motions. Errors and especially disturbances will be accumulated and affect the robot's odometry results. This is obvious for long distance travels or under adverse conditions.

There are mainly two kinds of error sources. One kind comes from the robot itself and another one is from outside. The former one includes the errors of robot's unequal wheel radii, unequal wheel distances, and misalignment of wheel angles and so on. These errors can be corrected since one can just measure the actual data from the real robot as precisely as possible and use them for the odometry calculation. Thus, (4.4) is extended to

$$\begin{bmatrix} x_g(t_1) \\ y_g(t_1) \\ \phi_g(t_1) \end{bmatrix} = \begin{bmatrix} x_g(t_0) \\ y_g(t_0) \\ \phi_g(t_0) \end{bmatrix} + \int_{t=t_0}^{t_1} \left( \boldsymbol{T}_{ex}(t_0) \cdot \begin{bmatrix} (r+e_{r_1})\omega_1(t_0) \\ (r+e_{r_2})\omega_2(t_0) \\ (r+e_{r_3})\omega_3(t_0) \end{bmatrix} \right) dt \tag{4.6}$$

with

$$\boldsymbol{T}_{ex}(t_0) = \begin{bmatrix} -\sin(\phi_g(t_0)+\alpha_1+e_{m_1}) & \cos(\phi_g(t_0)+\alpha_1+e_{m_1}) & R+e_{d_1} \\ -\sin(\phi_g(t_0)+\alpha_2+e_{m_2}) & \cos(\phi_g(t_0)+\alpha_2+e_{m_2}) & R+e_{d_2} \\ -\sin(\phi_g(t_0)+\alpha_3+e_{m_3}) & \cos(\phi_g(t_0)+\alpha_3+e_{m_3}) & R+e_{d_3} \end{bmatrix}^{-1}.$$

Here $e_{r_i}, e_{d_i}$ and $e_{m_i}(i=1,2,3)$, represent small deviations because of unequal wheel radii, unequal wheel distances and misalignment of wheel angles, respectively. A further investigation focusing on this is done at the institute which can be found in [19]. By this way, the error sources from robot itself can be reduced. This kind of odometry correction actually is a fine step of model modification which provides a more accurate odometry calculation.

However, during tests and experiments we noticed that the error sources from outside play a more important role, among them the wheel slippage is a key aspect. This means the robot moves less than the odometer counted because of slippage. Thus, this study also attempts to construct a slippage correction to the odometry calculation. One should know that there are many different kinds of global motions that can be performed by the used robot. Most important, its omnidirectional feature makes it possible to generate coupled motions, e.g., moving in $x$ direction in the global frame probably due to an actuation in $y$ direction together with a spinning in the robot's body frame. So, it is a troublesome task to build a uniform correction factor. However, if one changes the idea to work in the robot's local frame the question becomes easier. This is so, because either in the global or local frame all the movements are the combinations of two basic motion forms. These are the translational move ($x$ and $y$ directions) and the rotation. The important difference is if considered in a local frame, that one specific basic motion form always invokes the same wheel combination (also the same motors combination). Based on this principle, this research focuses on robot's body frame and for each of calculated steps a correction is made before projecting the motion to the global frame.

With this idea, a large number of experiments under different ground conditions need to be done since different kinds of grounds keep different slippery extents, although all of them only allow the same basic motion types. During one motion tests the other types of motion are isolated. After testing, one group of correction factors can be recommended for a specific ground, which in a discrete way can be governed by

$$\begin{bmatrix} \dot{x}_l^c(k) \\ \dot{y}_l^c(k) \\ \dot{\phi}_l^c(k) \end{bmatrix} = \boldsymbol{F}_c \cdot \boldsymbol{S}_l^{-1} \cdot \begin{bmatrix} (r+e_{r_1})\omega_1(k) \\ (r+e_{r_2})\omega_2(k) \\ (r+e_{r_3})\omega_3(k) \end{bmatrix} dt \tag{4.7}$$

where

$$\boldsymbol{F}_c = \begin{bmatrix} f_{c_1} & 0 & 0 \\ 0 & f_{c_2} & 0 \\ 0 & 0 & f_{c_3} \end{bmatrix}, \quad \boldsymbol{S}_l = \begin{bmatrix} -\sin(\alpha_1 + e_{m_1}) & \cos(\alpha_1 + e_{m_1}) & R + e_{d_1} \\ -\sin(\alpha_2 + e_{m_2}) & \cos(\alpha_2 + e_{m_2}) & R + e_{d_2} \\ -\sin(\alpha_3 + e_{m_3}) & \cos(\alpha_3 + e_{m_3}) & R + e_{d_3} \end{bmatrix}.$$

Here $\dot{x}_l^c(k), \dot{y}_l^c(k)$ and $\dot{\phi}_l^c(k)$ are the corrected local frame velocities of robot. From (4.7) one can see that the correction factors neither depend on step size nor the angular velocity of wheels. It is a diagonalized constant correction matrix without coupling from different dimensions. Such form of correction is what we pursue since it then only needs some simple tests to get the correction factors for a specific robot on a specific ground. This study uses a group of correction factors on a carpeted ground, in [19] a relatively smooth ground was also investigated.

Although this slippage correction improves the accuracy of robots odometry, one can not completely avoid all of the error sources. Therefore, one goes on to add other solutions, e.g., the following North Star measurements.

### *4.3.4 External North Star Measurements*

It is advantageous to add another sensor, typically an odometry independent sensor, to attain the robot pose. This kind of standalone data is also used for later performing data fusion in the Kalman filter.

#### 4.3.4.1 The North Star Navigation System

This study uses a novel external positioning sensor, the North Star system. It enables position and heading awareness in mobile robots, and can be applied in many position tracking applications in a variety of indoor situations. It breaks the 'line-of-sight' barrier of previous light beacon systems with simplicity and reliability. The concept of how this system does its measurements is illustrated in Fig. 4.6.

The North Star system is an infrared (IR) light based positioning device. When a projector (see Fig. 4.7) emits IR light which is reflected by the ceiling, the IR detector (see Fig. 4.8) which is mounted on the Robotino can receive the light signals and takes such signals as uniquely identifiable landmarks. By this, the robot can determine its relative position and direction. Of course the robot requires calibration and coordination before starting. This North Star positioning system is valid for several robots simultaneously by using different channels identified by varied frequency ranges. This also improves the positioning accuracy since it reduces the coordination errors compared to the situation of where each robot has its own global frame.

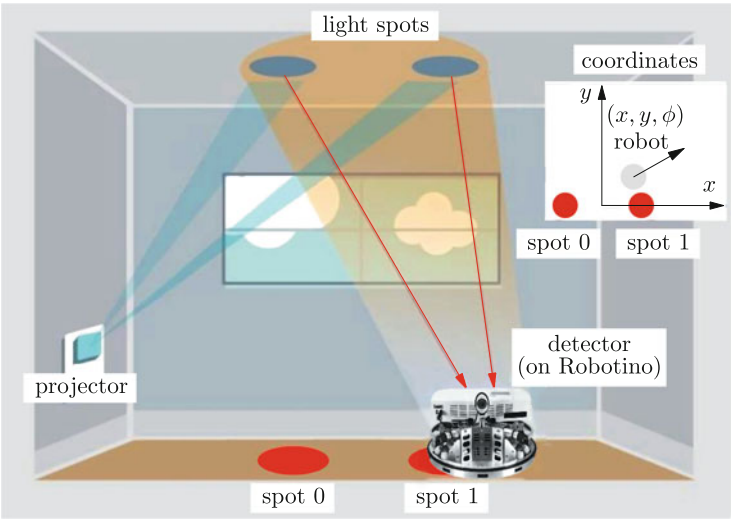



**Fig. 4.6** Illustration of North Star pose measurement

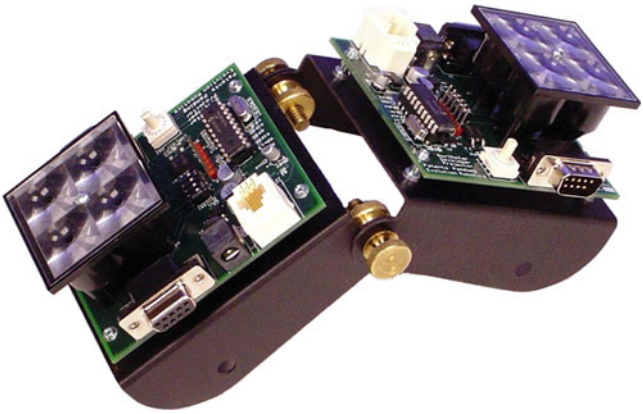


**Fig. 4.7** North Star projector (two emitter clusters)

#### 4.3.4.2 North Star Calibration

For odometry, there is no special calibration process since it only needs to refresh fast enough. However, the North Star needs calibration including building the coordinates before it can be used for measuring the robot pose. If several robots are used, they can share the same global coordinates. They need to gain the relationship between the changed pose and the changed infrared lights. Therefore, the calibration is done by performing a desired motion with exactly known position

**Fig. 4.8** North Star detector

and orientation changing traces. Meanwhile the North Star detector which is mounted on the Robotino records the infrared light changes corresponding to the robot motion. With triangulation calculus, the calibration purpose is achieved. The recording and calculus processes can be done by the North Star system itself, so one only needs to design a specific calibration motion and tell to the North Star.

## 4.4 Localization through Sensor Data Fusion in Extended Kalman Filters

Up to now, this study uses two methods to measure robot pose. Now the questions are, e.g., how to use the gained pose information for robot position control, how to weight the measurements from different sensors? From another side, we want to improve the position accuracy of robots while not putting too much burden on the external hardware. Thus, using methods to perform sensor data fusion will be a good choice.

### *4.4.1 Sensor Data Fusion*

Both the odometry and NS system provide position and orientation measurements of the robot. However, their accuracies are not very satisfactory even with further correction. Besides of the errors and uncertainties from odometry, the NS system will inevitably involve noises, too. For example, the North Star detector will exhibit nonlinear absolute localization responses and this nonlinearity will increase as the IR projector's light spots move away from the center of the detectors view field.

Additionally, position and orientation errors will also be produced when the detector is tilted from the plane of localization environment, e.g., the floor.

Due to above reasons, it is necessary to perform a pose correction based on the information from odometry and NS system. The most commonly used method in robotics to cope with this problem is the Kalman filter.

### 4.4.2 Kalman Filter and Extended Kalman Filter

In 1960, Rudolph E. Kalman published his famous paper [20] in which a recursive solution to the discrete-data filtering problem was described. Kalman filtering is a recursive process, rather than an electronic filter. In most cases, the Kalman filter is used for estimating the state of processes. A Kalman filter is an estimator for what is called linear-quadratic-problem, which is the problem of estimating the instantaneous 'state' of a linear dynamic system perturbed by white noise [21]. Kalman filters have been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This kind of estimation fits very well to the localization and position control for robot where the pose state of robot is a critical requirement. The robot obtains its poses through continually measuring by internal odometry and external North Star system, but neither the odometry nor North Star alone has a satisfactory accuracy since both of them contain noises. Therefore, it needs a method to weight the measurements and tries to fuse the information from both sides of measurements including the reduction of the noises affection.

Kalman filters address the general problem of trying to estimate the state $\boldsymbol{x} \in \Re^n$ of a discrete time controlled process that is governed by the linear difference equation

$$\boldsymbol{x}(k) = \boldsymbol{A} \cdot \boldsymbol{x}(k-1) + \boldsymbol{B} \cdot \boldsymbol{u}(k-1) + \boldsymbol{w}(k-1) \tag{4.8}$$

with a current measurement $\boldsymbol{z} \in \Re^m$ which is

$$\boldsymbol{z}(k) = \boldsymbol{H} \cdot \boldsymbol{x}(k) + \boldsymbol{v}(k). \tag{4.9}$$

Here $k$ is the time step, the $n \times n$ matrix $\boldsymbol{A}$ relates the state at the previous step $k-1$ to the state of at current step $k$ and depends on the odometry integration equation, see (4.4). Here $n$ is the dimensions of the system. In practical implementation $\boldsymbol{A}$ might change with each time step, e.g., the robot's odometry update in this study since the orientation $\phi_g$ in matrix $\boldsymbol{T}$ in (4.5) might change. The formulation of $\boldsymbol{A}$ for the Robotinos can be seen in the following nonlinear part. The $n \times l$ matrix $\boldsymbol{B}$ relates the optional control input $\boldsymbol{u} \in \Re^l$ to the state $\boldsymbol{x}$ while the $m \times n$ matrix $\boldsymbol{H}$ relates the state $\boldsymbol{x}$ to the measurement $\boldsymbol{z}$. For this study the state $\boldsymbol{x}$ includes the $x$ position, $y$ position and orientation of the robot which are the states concerned by our investigation. Please distinguish to the mechanical systems state vectors in

which usually the velocities are also included. Furthermore, here the odometry uses the internal encoders information to update, thus, there is no outside control input, i.e., $\boldsymbol{B}$ can vanish. The form of matrix $\boldsymbol{H}$ depends on the external sensor used for measurement, usually it contains the information like signal conversion, scaling, and so on. In this study, the external measurement is performed by the North Star system which directly provides the pose result thus $\boldsymbol{H}$ turns to be an unit matrix. The random variables $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ represent the process and measurement white noise, respectively. These noises satisfy the conditions of independent and according to normal probability distributions

$$\begin{aligned} p(\boldsymbol{w}) &\sim N(0, \boldsymbol{Q}), \\ p(\boldsymbol{v}) &\sim N(0, \boldsymbol{R}). \end{aligned} \tag{4.10}$$

Matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the process noise covariance and measurement noise covariance, respectively, and both of them might change. In our case, $\boldsymbol{Q}$ essentially depends on the noise of reading the motor speeds and $\boldsymbol{R}$ depends on the measurement noise from North Star. Robot's odometry contains an integration function. Qualifying Kalman filter to be used for the robot pose estimation needs to make a discretization of the odometry update as shown in (4.5).

The Kalman filter's recursive processes include not only the update of the differential equation (4.8), but also of the measurements and noises. This is in line with the concept of sensor data fusion since it needs to consider the information from different sensors. For our robot pose estimate case, one sensor is the internal odometry and another is the external North Star system. So, if the robot's pose update (estimate) is according to the Kalman recursive processes, it can fuse the useful information from both odometry and North Star. This is also the main reason why we use Kalman filters for the robot pose estimate.

The recursive processes of Kalman filters mainly consist of two parts, one part is the prediction and another part is the correction. The former one projects the current state estimate ahead over time and estimates the error covariance to obtain the a-priori estimate, noted as $\boldsymbol{x}(k)^-$ for step $k$ with the help of a 'super minus'. The correction part is responsible for incorporating a new measurement into the a-priori estimate so as to correct the projected estimate and obtain the a-posteriori estimate, noted as $\hat{\boldsymbol{x}}(k)$. A set of specific mathematical equations can be built to describe the prediction process by

$$\text{prediction} \begin{cases} \boldsymbol{x}(k)^- = \boldsymbol{A} \cdot \hat{\boldsymbol{x}}(k-1) + \boldsymbol{B} \cdot \boldsymbol{u}(k-1) \\ \boldsymbol{P}(k)^- = \boldsymbol{A} \cdot \boldsymbol{P}(k-1) \cdot \boldsymbol{A}^T + \boldsymbol{Q}(k-1) \end{cases} \tag{4.11}$$

and the correction by

$$\text{correction} \begin{cases} \boldsymbol{K}(k) = \boldsymbol{P}(k)^- \cdot \boldsymbol{H}^T \cdot (\boldsymbol{H} \cdot \boldsymbol{P}(k)^- \cdot \boldsymbol{H}^T + \boldsymbol{R}(k))^{-1} \\ \boldsymbol{P}(k) = (\boldsymbol{I} - \boldsymbol{K}(k) \cdot \boldsymbol{H}) \cdot \boldsymbol{P}(k)^- \\ \hat{\boldsymbol{x}}(k) = \boldsymbol{x}(k)^- + \boldsymbol{K}(k) \cdot (\boldsymbol{z}(k) - \boldsymbol{H} \cdot \boldsymbol{x}(k)^-) \end{cases} . \tag{4.12}$$

Except the notations of $\boldsymbol{P}$, $\boldsymbol{K}$ and $\boldsymbol{I}$, all others used in (4.11) and (4.12) have the same definitions as in (4.8), (4.9), and (4.10). Here $\boldsymbol{I}$ is an unit matrix, $\boldsymbol{P}(k)^-$ is the a-priori estimate error covariance which is defined by

$$\boldsymbol{P}(k)^- = \boldsymbol{E}\left[\boldsymbol{e}(k)^-(\boldsymbol{e}(k)^-)^T\right] \tag{4.13}$$

where $\boldsymbol{E}$ is the operation symbol for mathematical expectation and $\boldsymbol{e}(k)^- \equiv \boldsymbol{x}(k) - \boldsymbol{x}(k)^-$. The similar definition is given to $\boldsymbol{P}(k)$. However, the prediction and correction of $\boldsymbol{P}$ during the Kalman recursive processes are according to the respective equations in (4.11) and (4.12). The expressions in (4.11) and (4.12) for $\boldsymbol{P}$ are equivalent to the definition in (4.13) when the filter uses the optimal Kalman gain which is nearly always the case in practice, but they are in a clearer recursive form. Another important item in (4.12) is the so called Kalman gain or Kalman blending factor $\boldsymbol{K}$ which is used for guaranteeing to minimize the a-posteriori error covariance of $\boldsymbol{P}(k)$. Furthermore, the Kalman gain is also used for weighting the a-priori estimate and the measurement as shown in the a-posteriori equation. Specifically for this study, $\boldsymbol{K}$ is used for weighting the pose estimates from odometry and North Star. One form for the iteration of $\boldsymbol{K}$ is shown in (4.12), for more details see [22].

Originally, Kalman filters are designed for linear processes and the external measurement also should be linear. However, most practical problems are nonlinear, i.e., the system differential equation (4.8) becomes

$$\boldsymbol{x}(k) = \boldsymbol{f}(\boldsymbol{x}(k-1), \boldsymbol{u}(k-1), \boldsymbol{w}(k-1)). \tag{4.14}$$

The measurement equation (4.9) is then governed by

$$\boldsymbol{z}(k) = \boldsymbol{h}(\boldsymbol{x}(k), \boldsymbol{v}(k)). \tag{4.15}$$

Here $\boldsymbol{f}$ is the nonlinear function which relates the state at step $k-1$ to step $k$ while $\boldsymbol{h}$ is the nonlinear measurement function. Something akin to the Taylor series expansion, researchers linearize the estimation by using partial derivatives (Jacobian matrices) of the process and measurement. Thus, the extended Kalman filter (EKF) was developed, see details in [21]. Finally, a set of mathematical equations for the EKF can be described as

$$\text{prediction}\begin{cases} \boldsymbol{x}(k)^- = \boldsymbol{f}(\hat{\boldsymbol{x}}(k-1), \boldsymbol{u}(k-1), \boldsymbol{0}) \\ \boldsymbol{P}(k)^- = \bar{\boldsymbol{A}}(k)\cdot\boldsymbol{P}(k-1)\cdot\bar{\boldsymbol{A}}^T(k) \\ \qquad\qquad + \boldsymbol{W}(k)\cdot\boldsymbol{Q}(k-1)\cdot\boldsymbol{W}^T(k) \end{cases}, \tag{4.16}$$

$$\text{correction}\begin{cases} \boldsymbol{K}(k) = \boldsymbol{P}(k)^-\cdot\bar{\boldsymbol{H}}^T(k)\cdot(\bar{\boldsymbol{H}}(k)\cdot\boldsymbol{P}(k)^-\cdot\bar{\boldsymbol{H}}^T(k) \\ \qquad\qquad + \boldsymbol{V}(k)\cdot\boldsymbol{R}(k)\cdot\boldsymbol{V}^T(k))^{-1} \\ \boldsymbol{P}(k) = (\boldsymbol{I} - \boldsymbol{K}(k)\cdot\bar{\boldsymbol{H}}(k))\cdot\boldsymbol{P}(k)^- \\ \hat{\boldsymbol{x}}(k) = \boldsymbol{x}(k)^- + \boldsymbol{K}(k)\cdot(\boldsymbol{z}(k) - \boldsymbol{h}(\boldsymbol{x}(k)^-, \boldsymbol{0})) \end{cases}. \tag{4.17}$$

Here $\bar{\boldsymbol{A}}(k), \bar{\boldsymbol{H}}(k), \boldsymbol{W}(k)$ and $\boldsymbol{V}(k)$ are all Jacobian matrices containing partial derivatives

$$\bar{A}_{i,j}(k) = \frac{\partial f_i}{\partial x_j}(\hat{\boldsymbol{x}}(k-1), \boldsymbol{u}(k-1), \boldsymbol{0}), \tag{4.18}$$

$$\bar{H}_{i,j}(k) = \frac{\partial h_i}{\partial x_j}(\boldsymbol{x}(k)^-, \boldsymbol{0}), \tag{4.19}$$

$$W_{i,j}(k) = \frac{\partial f_i}{\partial w_j}(\hat{\boldsymbol{x}}(k-1), \boldsymbol{u}(k-1), \boldsymbol{0}), \tag{4.20}$$

$$V_{i,j}(k) = \frac{\partial h_i}{\partial v_j}(\boldsymbol{x}(k)^-, \boldsymbol{0}). \tag{4.21}$$

Usually the nonlinear function $\boldsymbol{f}$ is computationally expensive. Thus, researchers use approximation to the nonlinear processes and measurements.

### 4.4.3 Robotino Sensor Data Fusion for Localization

The investigated robot pose estimation is a nonlinear process thus it uses an extended Kalman filter. Using Taylor series expansion, we can perform the a-priori estimate according to (4.5). Additionally, if one writes the wheels angular velocities together with wheels radii and geometric structure of Robotino, then the a-priori estimate can be rearranged to

$$\begin{bmatrix} x_g(k)^- \\ y_g(k)^- \\ \phi_g(k)^- \end{bmatrix} = \begin{bmatrix} \hat{x}_g(k-1) \\ \hat{y}_g(k-1) \\ \hat{\phi}_g(k-1) \end{bmatrix}$$

$$+\Delta t \begin{bmatrix} \cos(\hat{\phi}_g(k-1)) & -\sin(\hat{\phi}_g(k-1)) & 0 \\ \sin(\hat{\phi}_g(k-1)) & \cos(\hat{\phi}_g(k-1)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_l^c(k-1) \\ \dot{y}_l^c(k-1) \\ \dot{\phi}_l^c(k-1) \end{bmatrix} \tag{4.22}$$

where $\dot{x}_l^c(k-1), \dot{y}_l^c(k-1)$ and $\dot{\phi}_l^c(k-1)$ are the last step's velocities given in the robot body frame. It is necessary to emphasize, here the used velocities in the local frame are already corrected according to (4.7). For the extended Kalman filter to perform the recursion for robot pose estimate one needs to know the functions $\boldsymbol{f}$ and $\boldsymbol{h}$ explicitly. Based on (4.22), one can write the three dimensions individually which yields

$$f_x = x_g(k)^- = \hat{x}_g(k-1) + \Delta t \cos(\hat{\phi}_g(k-1))\dot{x}_l^c(k-1) - \Delta t \sin(\hat{\phi}_g(k-1))\dot{y}_l^c(k-1) + \underbrace{0 \cdot \Delta t \dot{\phi}_l^c(k-1)}_{0}, \tag{4.23}$$

$$f_y = y_g(k)^- = \hat{y}_g(k-1) + \Delta t \sin(\hat{\phi}_g(k-1))\dot{x}_l^c(k-1) + \Delta t \cos(\hat{\phi}_g(k-1))\dot{y}_l^c(k-1) + \underbrace{0 \cdot \Delta t \dot{\phi}_l^c(k-1)}_{0}, \tag{4.24}$$

$$f_\phi = \phi_g(k)^- = \hat{\phi}_g(k-1) + \underbrace{0 \cdot \Delta t \dot{x}_l^c(k-1)}_{0} + \underbrace{0 \cdot \Delta t \dot{y}_l^c(k-1)}_{0} + \Delta t \dot{\phi}_l^c(k-1). \tag{4.25}$$

Next, for obtaining the a-priori estimate of error covariance $\boldsymbol{P}(k)^-$ one first needs to calculate Jacobian matrices $\bar{\boldsymbol{A}}(k)$ and $\boldsymbol{W}(k)$ according to (4.18) and (4.20) combined with the process functions of (4.23), (4.24), and (4.25). In our application this yields

$$\bar{\boldsymbol{A}}(k) = \left(\bar{A}_{i,j}(k)\right)_{3\times 3} = \begin{bmatrix} \frac{\partial f_x}{\partial \hat{x}_g(k-1)} & \frac{\partial f_x}{\partial \hat{y}_g(k-1)} & \frac{\partial f_x}{\partial \hat{\phi}_g(k-1)} \\ \frac{\partial f_y}{\partial \hat{x}_g(k-1)} & \frac{\partial f_y}{\partial \hat{y}_g(k-1)} & \frac{\partial f_y}{\partial \hat{\phi}_g(k-1)} \\ \frac{\partial f_\phi}{\partial \hat{x}_g(k-1)} & \frac{\partial f_\phi}{\partial \hat{y}_g(k-1)} & \frac{\partial f_\phi}{\partial \hat{\phi}_g(k-1)} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -\Delta t \sin(\hat{\phi}_g(k-1))\dot{x}_l^c(k-1) - \Delta t \cos(\hat{\phi}_g(k-1))\dot{y}_l^c(k-1) \\ 0 & 1 & \Delta t \cos(\hat{\phi}_g(k-1))\dot{x}_l^c(k-1) - \Delta t \sin(\hat{\phi}_g(k-1))\dot{y}_l^c(k-1) \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.26}$$

$$\boldsymbol{W}(k) = \left(W_{i,j}(k)\right)_{3\times 3} = \begin{bmatrix} \frac{\partial f_x}{\partial \dot{x}_l^c(k-1)} & \frac{\partial f_x}{\partial \dot{y}_l^c(k-1)} & \frac{\partial f_x}{\partial \dot{\phi}_l^c(k-1)} \\ \frac{\partial f_y}{\partial \dot{x}_l^c(k-1)} & \frac{\partial f_y}{\partial \dot{y}_l^c(k-1)} & \frac{\partial f_y}{\partial \dot{\phi}_l^c(k-1)} \\ \frac{\partial f_\phi}{\partial \dot{x}_l^c(k-1)} & \frac{\partial f_\phi}{\partial \dot{y}_l^c(k-1)} & \frac{\partial f_\phi}{\partial \dot{\phi}_l^c(k-1)} \end{bmatrix}$$

$$= \begin{bmatrix} \Delta t \cos(\hat{\phi}_g(k-1)) & -\Delta t \sin(\hat{\phi}_g(k-1)) & 0 \\ \Delta t \sin(\hat{\phi}_g(k-1)) & \Delta t \cos(\hat{\phi}_g(k-1)) & 0 \\ 0 & 0 & \Delta t \end{bmatrix}. \tag{4.27}$$

According to (4.20) the entries of $\boldsymbol{W}$ are the partial derivatives of $\boldsymbol{f}$ with respect to $\boldsymbol{w}$. However, practically $\boldsymbol{w}$ is always calculated through the variables of the noise sources. In robot navigation, if odometry is used as the internal pose sensor, then usually the considered noises are contained in robot's body velocities. In our case, originally the noise is mainly from robot's motor speed, however it is transferred and reflected on robot's velocity. Thus, in our applications, the entries of matrix $\boldsymbol{W}$ become the partial derivatives of $\boldsymbol{f}$ with respect to $\dot{x}_l^c(k-1), \dot{y}_l^c(k-1)$ and $\dot{\phi}_l^c(k-1)$. Of course here the noises don't include the noises which are already eliminated by odometry correction.

With $\bar{\boldsymbol{A}}(k)$ and $\boldsymbol{W}(k)$ one can calculate the prediction covariance

$$\boldsymbol{P}(k)^- = \bar{\boldsymbol{A}}(k) \cdot \boldsymbol{P}(k-1) \cdot \bar{\boldsymbol{A}}^T(k) + \boldsymbol{W}(k) \cdot \boldsymbol{Q}(k-1) \cdot \boldsymbol{W}^T(k). \tag{4.28}$$

Here $\boldsymbol{Q}(k-1)$ is the process noise covariance, see Sect. 4.4.5 for details. After the calculation of $\boldsymbol{P}(k)^-$ the extended Kalman filter gets into the processes of correction, i.e., the correction of the robot pose will be performed. For this the Kalman gain matrix must be calculated resulting in

$$\boldsymbol{K}(k) = \boldsymbol{P}(k)^- \cdot (\boldsymbol{P}(k)^- + \boldsymbol{R}(k))^{-1} \tag{4.29}$$

since both $\bar{\boldsymbol{H}}(k)$ and $\boldsymbol{V}(k)$ are unit matrices in this application, see Sect. 4.4.2 and (4.21). Here, the matrix $\boldsymbol{R}(k)$ is the measure noise covariance and it is one of the uncertainty sources. In Sect. 4.4.5 it will be determined.

Finally it performs the a-posteriori pose correction which will in each step be feed back to the desired pose for position control

$$\begin{bmatrix} \hat{x}_g(k) \\ \hat{y}_g(k) \\ \hat{\phi}_g(k) \end{bmatrix} = \begin{bmatrix} x_g(k)^- \\ y_g(k)^- \\ \phi_g(k)^- \end{bmatrix} + \boldsymbol{K}(k) \cdot \left( \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_\phi(k) \end{bmatrix} - \begin{bmatrix} x_g(k)^- \\ y_g(k)^- \\ \phi_g(k)^- \end{bmatrix} \right). \tag{4.30}$$

For the next step it updates the prediction covariance, here simplified to

$$\boldsymbol{P}(k) = (\boldsymbol{I} - \boldsymbol{K}(k)) \cdot \boldsymbol{P}(k)^-. \tag{4.31}$$

Here $z_x, z_y$ and $z_\phi$ are the three dimensions of pose information measured by the North Star system and $\boldsymbol{I}$ is a $3 \times 3$ unit matrix. From a mathematical point of view, in (4.30) $\boldsymbol{K}(k)$ is just a scaling factor to the two items, the $\boldsymbol{x}(k)^-$ and $\boldsymbol{z}(k)$. However, physically the Kalman gain weights the two sensors measurements, the odometry and North Star, by considering both of the sensors noises and uncertainties. Through this method, the sensors data are fused into the a-posteriori estimate of $\hat{\boldsymbol{x}}_g(k)$ having information from both odometry and North Star.

### 4.4.4 Data Flow Analysis

It is necessary to further clarify the data flows during this robot position control which is based on sensor data fusion in EKF. When the robot receives motion command for driving along one desired trajectory it immediately calls the algorithm which is listed in Algorithm 1. Here the trajectory consists of many control points $\boldsymbol{x}_d(i)$ in which $i$ is the index for the position points on the desired trajectory. This study tries to design the algorithm for general purpose. Specifically speaking, if the position control purpose is only for the robot to reach a final position and it doesn't care about the robot's trajectory, then one can directly give the final desired point $\boldsymbol{x}_d(\text{final})$ to the robot. Otherwise, give the position points on the desired trajectory one by one to the robot which means the robot should traverse all of these points so as to track this trajectory. The more position points are given, the more accurate trajectory will be performed by the robot, but of course within the restrictions of computational power and hardware.

Algorithm 1 is the pseudo-code for our experiments. In addition to the EKF algorithm itself, here the processes for position control are also shown which are directly oriented to the implementation and involved hardware.

**Algorithm 1.** Robot position control based on sensor data fusion in EKF

/* initialize: get system information and initial conditions (including robot initial pose $\hat{\boldsymbol{x}}_g(0)$, read in initial motor speed $[\omega_{m_1}(0), \omega_{m_2}(0), \omega_{m_3}(0)]$ and EKF start up conditions $\boldsymbol{P}(0)$, $\boldsymbol{Q}(0), \boldsymbol{R}(0)$), specify time step $\Delta t$, step $k = 0$, control point notation $i = 1$, calculate initial deviation $\Delta\boldsymbol{x}_g(0)$, define tolerance threshold $\epsilon$ for position error*/
**while** $\Delta\boldsymbol{x}_g(k) > \epsilon$ or $i <$ final **do**
  **if** $\Delta\boldsymbol{x}_g(k) \leq \epsilon$ **then**
    $i = i + 1$
    **input**: the new control point $\boldsymbol{x}_d(i)$ from the desired trajectory
    calculate the new pose deviation $\Delta\boldsymbol{x}_g(k) = \boldsymbol{x}_d(i) - \hat{\boldsymbol{x}}_g(k)$
**end if**
drive (or compensate) the robot pose due to $\Delta\boldsymbol{x}_g(k)$
update EKF step index $k = k + 1$
convert last step's motor speeds to wheels speeds $[\omega_1(k-1), \omega_2(k-1), \omega_3(k-1)]$
calculate robot velocity in body coordinates, including odometry correction due to (4.7)
perform the a-priori pose estimation $\boldsymbol{x}_g(k)^-$ by odometry, (4.22), and store the current robot velocity $\dot{\boldsymbol{x}}_g(k)$
calculate Jacobian matrices $\bar{\boldsymbol{A}}(k)$ and $\boldsymbol{W}(k)$ by (4.26) and (4.27)
**if** $k = 1$ **then**
  use $\boldsymbol{Q}(0)$ as the covariance of process noise
**else**
  read in last step's motor speeds, update the covariance of process noise $\boldsymbol{Q}(k-1)$ by (4.32) and (4.33)
**end if**
calculate prediction covariance $\boldsymbol{P}(k)^-$, (4.28)
update the covariance of measurement noise $\boldsymbol{R}(k)$, (4.34)
calculate Kalman gain matrix $\boldsymbol{K}(k)$, (4.29)
read in the external measurement $\boldsymbol{z}(k)$ from North Star system

(continued)

23: perform the a-posteriori pose correction, get $\hat{\hat{\boldsymbol{x}}}_g(k)$ by (4.30)
24: compare the EKF estimated pose $\hat{\hat{\boldsymbol{x}}}_g(k)$ with the control point $\boldsymbol{x}_d(i)$, refresh deviation $\Delta\boldsymbol{x}_g(k)$
25: update prediction covariance $\boldsymbol{P}(k)$ for next step, (4.31)
26: get the three motors current speeds $[\omega_{m_1}(k), \omega_{m_2}(k), \omega_{m_3}(k)]$
27: **end while**
28: **output**: final pose $\hat{\hat{\boldsymbol{x}}}_g(k)$ and velocity $\dot{\boldsymbol{x}}_g(k)$ (the information provided for mechanical PSO's iteration in our upper level algorithm)

### 4.4.5 Uncertainty Analysis

There are many kinds of uncertainties during the data fusion for robot localization and position control, some of them are known and some are not. In this study, basically the uncertainties come from two parts. One part is the robot and its motion, i.e., the values of physical parameters, uneven terrain, slippery floor and so on. Another kind of uncertainty is from the sensor measurement data. There are no specific methods (formulae or procedures) to determine the covariance matrices $\boldsymbol{Q}(k)$ and $\boldsymbol{R}(k)$. The relatively effective way is by trials and relying on experience.

First, it is important to choose a reasonable initial value. Here, the process noise covariance $\boldsymbol{Q}(k)$ is mainly due to the robot's velocity noise which is essentially from the robot's motors. As such, it is possible to get the basic noises information from the DC motors data sheet and some simple tests. Such tests can be performed by giving a planned constant motor speed value to one specific motor, and let the motor run for a long time until we consider it is running stable. Then, we let the encoder read back the current actual value of motor speed which will be used for odometry calculation. By comparing the read in actual value and the planned value one can get an idea about the noise range of the motor. With the drivetrain relationships, the noise from the motors can be converted to the global pose noise which is formulated as

$$\boldsymbol{Q}(k) = c_p \hat{\boldsymbol{Q}}(k) \tag{4.32}$$

with

$$\begin{aligned}\hat{\boldsymbol{Q}}(k) &= \boldsymbol{T}_{\text{trans}} \cdot \begin{bmatrix} \text{cov}(\boldsymbol{N}_1(k), \boldsymbol{N}_1(k)) & 0 & 0 \\ 0 & \text{cov}(\boldsymbol{N}_2(k), \boldsymbol{N}_2(k)) & 0 \\ 0 & 0 & \text{cov}(\boldsymbol{N}_3(k), \boldsymbol{N}_3(k)) \end{bmatrix} \\ &= \boldsymbol{T}_{\text{trans}} \cdot \begin{bmatrix} \text{D}(\boldsymbol{N}_1(k)) & 0 & 0 \\ 0 & \text{D}(\boldsymbol{N}_2(k)) & 0 \\ 0 & 0 & \text{D}(\boldsymbol{N}_3(k)) \end{bmatrix}\end{aligned} \tag{4.33}$$

where $c_p$ is a scaling factor, $\boldsymbol{N}_i(k)(i=1,2,3)$ is a vector which contains the $i$-th motor's process noises sequence up to the current step while $\mathrm{cov}(\boldsymbol{N}_i(k),\boldsymbol{N}_i(k))$ is the $i$-th motor's noise covariance. The scalers $\mathrm{D}(\boldsymbol{N}_i(k))$ represent the variance and mathematically $\mathrm{cov}(\boldsymbol{N}_i(k),\boldsymbol{N}_i(k))=\mathrm{D}(\boldsymbol{N}_i(k))$. Here $\boldsymbol{T}_{\mathrm{trans}}$ is a transformation matrix which includes the information of time step $\Delta t$ and the conversion from motor speeds to robot's global velocities.

We still left the question of how to get the noises vector $\boldsymbol{N}_i(k)$. By simple tests one already knows the range of motor noise, this can be used for calculating the initial value $\boldsymbol{Q}(k)$. However, the motor noise (velocity noise) counted into the robot odometry calculation is changing during the robot's movement. So the more precise way is to update it in each step. This is done by comparing the real time read in motor speed with the nominal motor speed which is computed by the current pose deviation $\Delta\boldsymbol{x}_g(k)$ and inverse kinematics. Here the nominal motor speed actually is the command value one gives to the motor controller. By this the motor process noises can be estimated. Worth to be noticed, here it only obtains the execution process noises of the motors, and the encoders are not possible to count the uncertainties like the wheel slippage, so this part of noise without consideration of slippage noise. Wheels slippage noise partially is handled (eliminated) by the odometry correction, and the residual part is transmitted finally to the robot velocity noise together with motor execution noise. Other noises and uncertainties can be considered in the scaling factor $c_p$ which is convenient to be adjusted.

From (4.32) and (4.33) one can see that the motor covariances are assumed to be diagonal, however its contribution to the EKF pose estimation is not in a diagonal way. One reason is its transformation matrix $\boldsymbol{T}_{\mathrm{trans}}$, and another reason is from the second item of (4.28) where the Jacobian matrix $\boldsymbol{W}$ and its transpose couple the different dimensions of $\boldsymbol{Q}(k)$. This is actually physically reasonable.

Akin to the process noise covariance, the measure noise covariance in this application is governed by

$$\boldsymbol{R}(k)=c_m\begin{bmatrix}\mathrm{cov}(\boldsymbol{X}_1(k),\boldsymbol{X}_1(k)) & \mathrm{cov}(\boldsymbol{X}_1(k),\boldsymbol{X}_2(k)) & \mathrm{cov}(\boldsymbol{X}_1(k),\boldsymbol{X}_3(k))\\ \mathrm{cov}(\boldsymbol{X}_2(k),\boldsymbol{X}_1(k)) & \mathrm{cov}(\boldsymbol{X}_2(k),\boldsymbol{X}_2(k)) & \mathrm{cov}(\boldsymbol{X}_2(k),\boldsymbol{X}_3(k))\\ \mathrm{cov}(\boldsymbol{X}_3(k),\boldsymbol{X}_1(k)) & \mathrm{cov}(\boldsymbol{X}_3(k),\boldsymbol{X}_2(k)) & \mathrm{cov}(\boldsymbol{X}_3(k),\boldsymbol{X}_3(k))\end{bmatrix} \tag{4.34}$$

where

$$\begin{aligned}\boldsymbol{X}_1(k)&=\{\Delta x_g(1),\Delta x_g(2),\cdots,\Delta x_g(k)\},\\ \boldsymbol{X}_2(k)&=\{\Delta y_g(1),\Delta y_g(2),\cdots,\Delta y_g(k)\},\\ \boldsymbol{X}_3(k)&=\{\Delta\phi_g(1),\Delta\phi_g(2),\cdots,\Delta\phi_g(k)\},\end{aligned}$$

are the corresponding noise sequences of North Star measured $x$, $y$ positions and $\phi$ orientations, $c_m$ is a scaling factor. Here $\boldsymbol{R}(k)$ directly contributes in a coupled way (non-diagonal). Now the question is left how to get the noise (error) sequences $[\Delta x_g(i),\Delta y_g(i),\Delta\phi_g(i)],(i=1,2,\cdots,k)$, which can basically describe the noise

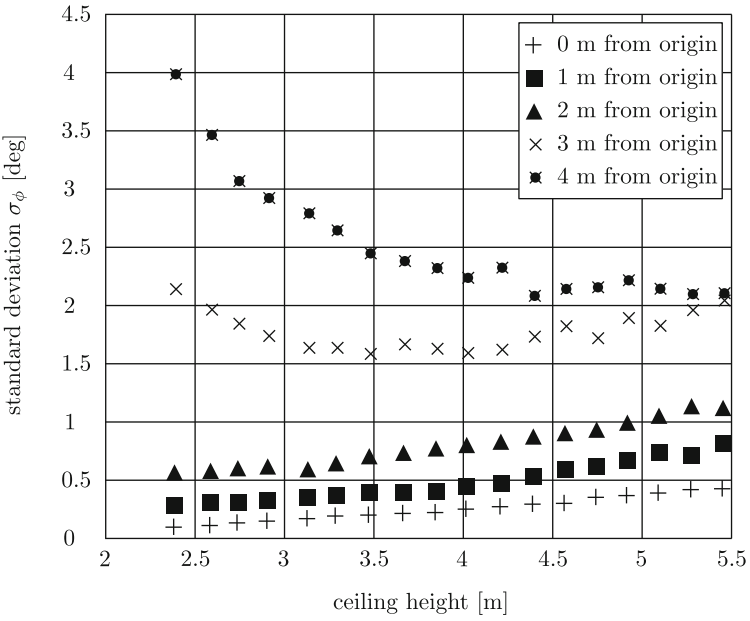



**Fig. 4.9** The North Star orientation noise when used 4 LEDs in projector (data from information of Evolution Robotics, Inc.)

during the North Star measurements. One gets help from the North Star data sheets. Here one takes the orientation noise as an example, see Fig. 4.9.

Figure 4.9 shows the standard deviation of the orientation errors of the North Star system with different ceiling height and different distances from the origin. Usually, in a specific experiment room with a fixed ceiling height, the robot moving in different locations gains different position and orientation noises, so the measure noise covariance is changing. As we can see from Fig. 4.9, the standard deviation of orientation error changes approximately from $0.15^\circ$ to $2.8^\circ$ as the distance changes from 0 to 4 m in a room with a ceiling around 3 m high. Obviously, from Fig. 4.9 one can see that this is a nonlinear relationship. However, such trends can be represented by a polynomial interpolated curve. The input of this curve is the current distance from the robot to the origin, in this case we get it from $\boldsymbol{x}_g(k)^-$. By this the sequence of measure noise for robot orientation is obtained. With the current noises sequence, one can calculate the noises covariance easily. Similarly, the covariances for position noises can be obtained, too.

Both $\boldsymbol{Q}(k)$ and $\boldsymbol{R}(k)$ are updated in each step during the EKF recursion. This is different to other traditional EKF applications where usually the covariance matrices for process noise and measure noise are fixed. Refreshing them in each step gives us a relatively closer value to the true noises.

## 4.5 Experiments and Results Analysis

In this section the previously described method is verified on a real physical Robotino robot, and then the obtained results are compared to the results from motions without performing sensor data fusion, including no feedback and only odometry feedback position control. Additionally, the method presented by this study is also briefly compared to the traditional navigation method, the simultaneously localization and mapping (SLAM).

### *4.5.1 Experimental Setup*

#### 4.5.1.1 Environment Requirements

The experiments are performed in an indoor environment which has a carpeted, flat ground. The room ceiling is around 3 m high and it's better to keep few objects in the room because we want to reduce the reflecting interferences from objects. Furthermore, it is also necessary to keep away as good as possible light sources including the daylight. A representative experimental environment is illustrated in Fig. 4.10.

#### 4.5.1.2 Experiment Arrangement

Three groups of experiments are designed in which one group is done without position feedback control and the other two groups are all performed with closed loop feedback control. The open loop group uses an input-state equations-output style. The second group of experiments only uses the corrected odometry for measurement, and the measured results are feed back for position control. For comparisons, the third group is performed with sensor data fusion using an extended Kalman filter where the pose information is not only from the odometry but also from the North Star system. Then, the estimated pose is used for the position feedback control. Each group of experiments is organized with six kinds of motions and each motion gets 20 runs. Complex motions are combined from such basic types. The motion types and experiment dimensions are illustrated in Fig. 4.11.

The six kinds of motions are: M1 move forward 1 m in global $x$ direction, M2 rotate with a half circle, M3 move diagonal with ($\Delta 1m$, $\Delta 1m$), M4 spin at the original place $360^{\circ}$, M5 move in $-y$ direction with 1 m and M6 again move forward 1 m in $x$ direction.

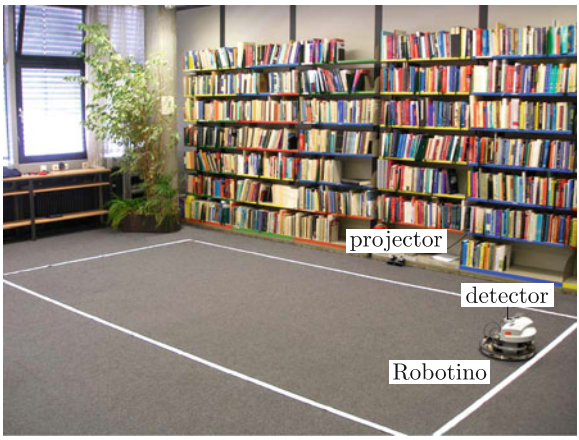



**Fig. 4.10** Environment for localization and position control experiments

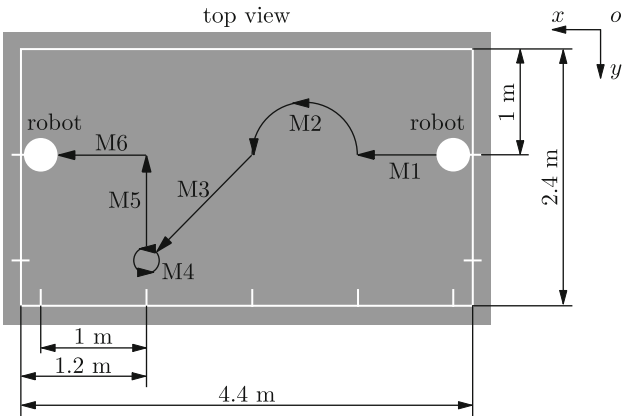



**Fig. 4.11** Motion types

## *4.5.2 Localization and Position Control Experiments*

Through the designed three groups of experiments this study aims to investigate how much the described approaches can improve localization and position control quality.

### 4.5.2.1 Open Loop Position Control

Open loop position control means the robots are moving by assigning specific velocities and running time, there are no actual pose measurement and no state

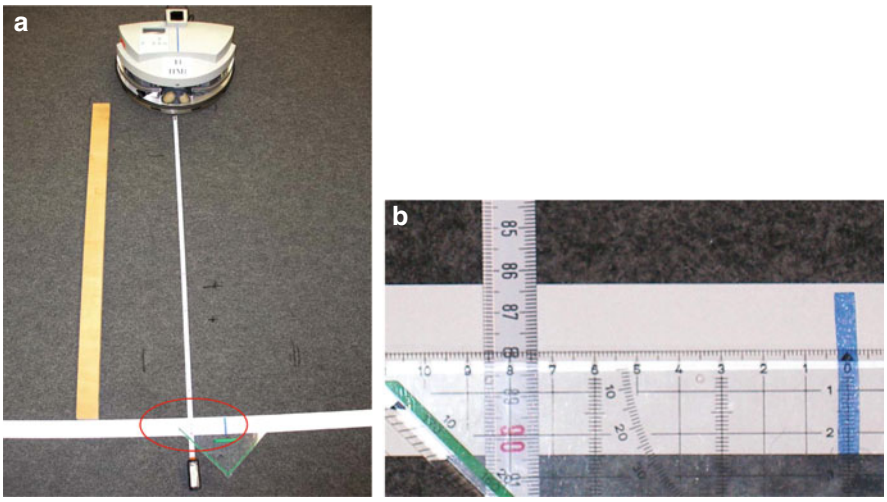



**Fig. 4.12** Open loop position control for motion M1

feedback. One typical run for moving forward 1 m in $x$ direction of the open loop position control is shown in Fig. 4.12.

#### 4.5.2.2 Odometry Based Feedback Position Control

The second group of experiments uses only the measurements from the corrected odometry. This is a single sensor relied position control method. The parameters used in the odometry calculation for error sources are

$$\begin{aligned}\boldsymbol{e}_r &= [e_{r_1}, e_{r_2}, e_{r_3}]^T = [-0.16\text{mm}, 0.12\text{mm}, -0.03\text{mm}]^T,\\ \boldsymbol{e}_d &= [e_{d_1}, e_{d_2}, e_{d_3}]^T = [-2\text{mm}, 2\text{mm}, -10\text{mm}]^T,\\ \boldsymbol{e}_m &= [e_{m_1}, e_{m_2}, e_{m_3}]^T = [3^\circ, 2^\circ, 2^\circ]^T,\end{aligned}$$

the nominal values are

$$r = 40\text{mm}, \qquad R = 135\text{mm},$$

and for the odometry correction matrix we use $\boldsymbol{F}_c = \boldsymbol{diag}(0.77\ 0.77\ 1.01)$. A more complete description of the corrected odometry can be found in [19].

#### 4.5.2.3 Localization and Closed Loop Position Control with Odometry and North Star Data Fusion in EKF

In the third group of experiments one needs information from both odometry and North Star. The obtained results are then fused by the extended Kalman filter. Through

this the improved pose information can be obtained. The used initial covariances of process and measure noise are $\boldsymbol{Q}(0) = \boldsymbol{diag}(10^{-4},\ 10^{-4},\ (3 \times \pi/180)^2)$ and $\boldsymbol{R}(0) = \boldsymbol{diag}(6.76 \times 10^{-4},\ 6.76\times\ 10^{-4},\ (7 \times \pi/180)^2)$, respectively. They all have the units $[\mathrm{m}^2,\ \mathrm{m}^2,\ \mathrm{rad}^2]$. The odometry has a start up error of 1 cm for the $x$ and $y$ directions and 3° for the orientation error. Similarly, the North Star initial error is estimated as $[2.60\mathrm{cm},\ 2.60\mathrm{cm},\ 7^\circ]$ where a bigger orientation error is used since the North Star system has a worse measurement for the orientation at the beginning although the robot is near to the origin at that time and according to Fig. 4.9 this value is only around 0.2°. The initial estimate error covariance is $\boldsymbol{P}(0) = \boldsymbol{diag}(10^{-4}\mathrm{m}^2,\ 10^{-4}\mathrm{m}^2,\ (\pi/60)^2\,\mathrm{rad}^2)$ and the time step is $\Delta t = 0.01\mathrm{s}$. For the North Star noises fitting curves we tried 3-rd order, 4-th order and 5-th order polynomials, and we find the 4-th order polynomial is reasonable which for $x$, $y$ directions and orientation noises are

$$\begin{aligned}
\sigma_x &= -0.0010dis^4 + 0.0128dis^3 - 0.0285dis^2 + 0.0265dis + 0.0003,\\
\sigma_y &= -0.0007dis^4 + 0.0054dis^3 - 0.0098dis^2 + 0.0066dis + 0.0025,\\
\sigma_\phi &= -0.0729dis^4 + 0.5625dis^3 - 1.1271dis^2 + 0.8375dis + 0.1500.
\end{aligned}$$

Here *dis* represents the current distance from the origin to the robot. With above North Star noise expressions, one can update the noise covariance $\boldsymbol{R}(k)$ for each step. The odometry related values used in this group of experiments are exactly the same as in Sect. 4.5.2.2 for odometry feedback position control.

### 4.5.3 Result Analysis and Comparisons

Each of the six motions is performed with 20 runs and we compare their average values for the three different groups. The statistical results can be seen in Fig. 4.13.

One can see that the positioning accuracy is improved by using odometry or EKF based localization. Taking motion M1 as an example, in the odometry feedback case, the final position error is 0.3 cm in average. This is further improved by the EKF based closed loop position control where the localization error is only 0.1 cm in average. It is assumed that the controller can drive exactly according to the feedback localized poses which means the final 0.3 cm or 0.1 cm errors come from the localization, rather than from the controller. Thus, the experiment results can directly represent the localization quality.

The errors are increasing as the robot is moving away from the origin, i.e., this leads to bigger deviations for long distance movements. This can be seen from motions M1 and M6 in Fig. 4.13. For these two motions, although they perform the same behavior, the errors from all of the three groups of experiments are increased. However, the EKF based one shows the highest robustness since it keeps the smallest deterioration rate.

The robot orientation result is still not satisfactory since the orientation error is more sensitive compared to the translational error because of the omnidirectional feature, see e.g., the results of M4. From motion M4 one also can see that the

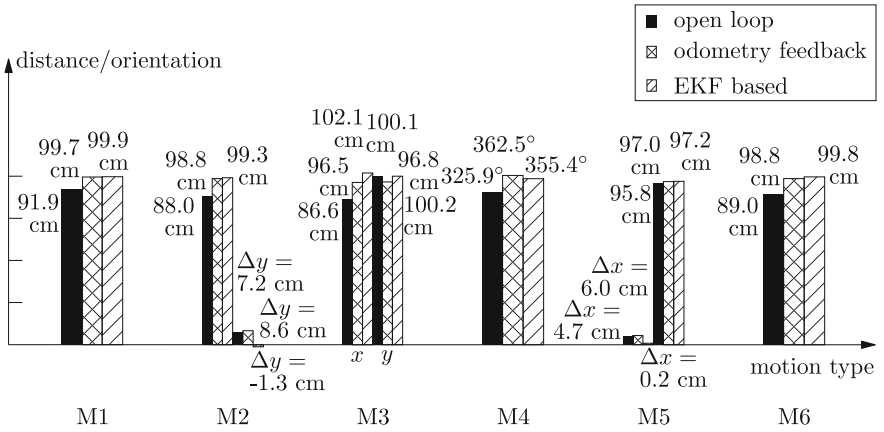



**Fig. 4.13** Statistical results of open loop, odometry based closed loop and EKF based closed loop position control experiments

'odometry only' feedback control there even obtains a better result than the EKF based one. This is because the odometry, especially our corrected odometry, has a good localization quality in the spinning motion, whereas the North Star system has a worse orientation measurement especially when the robot moves away from its origin. Basically, the EKF based feedback control can further improve the translational positioning accuracy compared to the odometry based case, however the orientation accuracy is slightly worse than in the case where only odometry is used.

In the following some comparisons to a traditional method are performed. In most of the cases, researchers use simultaneously localization and mapping (SLAM) method. It actually has become a de-facto standard for robots navigation and related tasks. The so called SLAM is a technique used by robots and autonomous vehicles to build up a map within an unknown environment (without a priori knowledge) or to update a map within a known environment (with a priori knowledge from a given map) while at the same time keeping track of their current locations. However, this method is computational costly, especially for the unknown environment mapping. It usually needs many vision techniques and image processing knowledge which are not suitable for our application since the calculation ability on the robot itself is very restricted.

As such, this investigation avoids the SLAM method, at least avoids the mapping part. With PSO's powerful search ability this becomes possible. The mechanical PSO is guiding the robot (generating trajectories) and the robot is handling some things locally such as, e.g., obstacle avoidance and localization. By this method, it is neither necessary to obtain the environment information precisely nor fully. With the help of corrected odometry or corrected odometry and North Star data fusion in the extended Kalman filter, the localization quality is improved.

## 4.6 Conclusions

This contribution investigates the robot localization. For this purpose, methods based on corrected odometry and North Star measurements are developed. Then, improve positioning accuracy by sensor data fusion in the extended Kalman filters. Two measurement channels for EKF data fusion are the corrected odometer and the external North Star system. The data flows and uncertainties during the localization process are analyzed. Finally, the developed methods are verified by robot position control experiments with one group of open loop runs, one group of 'odometry only' closed loop feedback runs and one group of EKF based closed loop feedback runs. The experimental results carried out by an omnidirectional robot on a carpet ground under six kinds of motions demonstrate the feasibility of odometry based or EKF based closed loop feedback control for improving robot positioning accuracy. Due to the localized information, the robot pose control can be performed successfully. For an application where the pose accuracy is not a strict requirement, the control based on the corrected odometry is a good and sufficient choice. For the case where higher accuracy is a necessity, and the environment allows to perform sensor data fusion, then the EKF based pose control will be a nicer choice although with limitation for improving the orientation accuracy. From the comparisons, one can see that the proposed techniques show promising results when compared to the no localization case.

**Acknowledgement** The second author gratefully acknowledges the Chinese Scholarship Council (CSC) for supporting his research in Germany. This work is supported in part also by the Cluster of Excellence Simulation Technology (SimTech) in Stuttgart, Germany.

## References

1. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. MIT Press, Cambridge
2. Yang Y (2008) Probabilistic path planning with extended local planners. Ph.D. thesis, Purdue University
3. Ferdaus SN (2008) A topological approach to online autonomous map building for mobile robot navigation. Master thesis, Memorial University of Newfoundland
4. McLurkin J (2009) Measuring the accuracy of distributed algorithms on multi-robot systems with dynamic network topologies. Distrib Auton Robot Syst 8:15–26, Springer
5. Murillo AC, Guerrero JJ, Sagüés C (2007) Topological and metric robot localization through computer vision techniques. In: Proceedings of 2007 IEEE workshop from features to actions: unifying perspectives in computational and robot vision held with IEEE international conference on robotics and automation, Rome, pp 79–85
6. Cao MM (2009) Dynamic behavior path planning of mobile robot based on fuzzy logic control (in Chinese). Master thesis, Beijing Jiaotong University
7. Maaref H, Barret C (2002) Sensor-based navigation of a mobile robot in an indoor environment. Robot Auton Syst 38(1):1–18
8. Liu Y, Zhu JJ, Williams RL, Wu JH (2008) Omni-directional mobile robot controller based on trajectory linearization. Robot Auton Syst 56(5):461–479

9. Wu JH (2005) Dynamic path planning of an omni-directional robot in a dynamic environment. Ph.D. thesis, Ohio University
10. Caglioti V, Citterio A, Fossati A (2006) Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach. In: Proceedings of the 2006 IEEE workshop on distributed intelligent systems: collective intelligence and its applications, Prague, pp 25–30
11. Fox D, Burgard W, Kruppa H, Thrun S (2000) A probabilistic approach to collaborative multi-robot localization. Auton Robot 8(3):325–344
12. Howard A, Matark M, Sukhatme G (2002) Localization for mobile robot teams using maximum likelihood estimation. In: Proceedings of 2002 IEEE/RSJ international conference on intelligent robots and systems, vol 1. Lausanne, pp 434–439
13. Madhavan R, Fregene K, Parker LE (2004) Distributed cooperative outdoor multirobot localization and mapping. Auton Robot 17(1):23–39
14. Roumeliotis SI, Bekey GA (2002) Distributed multirobot localization. IEEE Trans Robot Autom 18(5):781–795
15. Eberhard P, Tang Q (2011) Particle swarm Optimization used for mechanism design and guidance of swarm mobile robots. In: Olsson AE (ed) Particle swarm optimization: theory, techniques, and applications. Nova Science, New York, pp 193–225
16. Tang Q, Eberhard P (2011) Cooperative motion of swarm mobile robots based on Particle Swarm Optimization and multibody system dynamics. Mech Based Des Struct Mach 39(2):179–193
17. Chen CT (1999) Linear system theory and design, 3rd edn. Oxford University Press, Cary
18. Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, Hoboken
19. Munir R (2012) Odometry error propagation and correction for one kind of omnidirectional mobile robots. Student thesis STUD-369, Institute of Engineering and Computational Mechanics, University of Stuttgart
20. Kalman RE (1960) A new approach to linear filtering and prediction problems. Basic Eng 82(1):35–45
21. Grewal MS, Andrews AP (2008) Kalman filtering: theory and practice using MATLAB, 3rd edn. Wiley, Hoboken
22. Jacobs OLR (1993) Introduction to control theory, 2nd edn. Oxford University Press, Cary
23. Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybernet 4(2):100–107
24. Stentz A (1994) Optimal and efficient path planning for partially-known environments. Proc IEEE Int Conf Robot Autom 4:3310–3317, San Diego
25. Borenstein J, Koren Y (1991) The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Trans Robot Autom 7(3):278–288
26. Khatib O (1986) Real time obstacle avoidance for manipulators and mobile robots. Int J Robot Res 5(1):90–98
27. Ulrich I, Borenstein J (1998) VFH+: reliable obstacle avoidance for fast mobile robots. Proc IEEE Int Conf Robot Autom 2:1572–1577, Leuven
28. Ulrich I, Borenstein J (2000) VFH*: local obstacle avoidance with look-ahead verification. Proc IEEE Int Conf Robot Autom 3:2505–2511, San Francisco

# Chapter 5
# Modelling and Control of Infinite-Dimensional Mechanical Systems: A Port-Hamiltonian Approach

**Markus Schöberl and Andreas Siuka**

**Abstract** We consider a port-Hamiltonian representation for infinite-dimensional systems described by partial differential equations. Then the control by interconnection method is applied, by using a finite-dimensional controller system interacting via an energy port at the boundary of the infinite-dimensional system. This will be demonstrated by means of a heavy chain system, modelled as a partial differential equation. Furthermore, we sketch the stability proof in the infinite-dimensional setting. To motivate for the presented ideas we recapitulate the well-known concepts for finite-dimensional systems as well, but mainly as a starting point for the discussion of the infinite setting.

## 5.1 Introduction

Port-controlled Hamiltonian systems with Dissipation (PCHD systems) are very popular in system theory since this special representation of the system equations highlights the physics used in a remarkable way. In many applications the Hamiltonian corresponds to the total energy of the system and the PCHD structure reveals how the energy is conserved and/or dissipated by corresponding physical elements. This representation is also widely utilized in control, mainly with regard to concepts where the control strategy is based on the analysis of the power flows. These investigations have been extensively exploited in the case of finite dimensional systems described by ordinary differential equations (ODEs), see [1] and references therein. A key advantage of PCHD systems in the finite-dimensional case is the fact that the system structure is directly linked to stability theory, since it can be deduced that the equilibrium corresponding to a minimum of the positive definite

M. Schöberl (✉) • A. Siuka
Institute of Automatic Control and Control Systems Technology, Johannes Kepler University, Linz, Austria
e-mail: markus.schoeberl@jku.at; andreas.siuka@jku.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_5, 

Hamiltonian is stable in the sense of Lyapunov. If asymptotic stability cannot be concluded by using the Hamiltonian it is common to make use of La-Salle's Invariance principle, see e.g. [2].

The Hamiltonian picture mentioned above is however not limited to model finite-dimensional systems but it is also adequate for the infinite-dimensional case, e.g. systems described by partial differential equations (PDEs). The PCHD concept in the PDE case is not unique and several approaches exist in the literature, see for example [3–8]. The approach used in this contribution is based on a setting also used in [9] which has been adapted for our purposes (non-trivial boundary conditions and introducing in- and outputs) in [7, 8, 10, 11]. The advantage of a Hamiltonian representation also for systems described by PDEs is the fact that many ideas and concepts well-known from the ODE case can be copied and/or adapted. However, a main drawback is the loss of the direct link to stability theory – this is much more involved than in the finite-dimensional case, and tools such as semi-group theory become indispensable, see e.g. [12, 13].

In this paper we want to analyze the control by interconnection concept, where the focus is laid on the PDE case. To motivate for the ideas we recapitulate the ODE scheme first, where we present the example of a pendulum with end-mass whose suspension point can move in the horizontal direction. This system with two degrees of freedom (pendulum angle and position of suspension point) will be modeled in a Hamiltonian fashion and a controller will be designed to stabilize the system in the downward position with a prescribed position of the suspension point. It is easy to show that the closed loop system structure which will be again Hamiltonian leads directly to the stability arguments. Then to shift to the PDE case we will replace the pendulum by a chain and apply the same controller design procedure adapted to the PDE case. It will be then shown that the stability proof is much harder compared to the ODE case. Beside these two examples the theoretical concepts will be touched but not worked out in detail, concerning the control of infinite-dimensional systems with the control by interconnection technique based on our system representation the reader is referred to [10, 11] or for more details concerning the ODE case to [1] and references therein.

This paper is organized as follows. In Sect. 5.2 we present the well-known case of Port-controlled Hamiltonian Systems in the case of ordinary differential equations. Then the concept of Casimir functions and the control by interconnection technique are recapitulated. The pendulum system with an end-mass is used to demonstrate the presented concepts. The third section is then focused on the PDE case where the same considerations as before are carried out but adapted to work also for infinite-dimensional systems. Instead of the pendulum system a heavy chain system modeled as a partial differential equation will be used. A sketch of the stability proof is presented subsequently. Finally, in the fourth section a summary and a discussion is given.

## 5.2 The Finite-Dimensional Case

In this section we will analyze Port-controlled Hamiltonian systems with Dissipation (PCHD systems). We will sketch the well known system representation as well as the control by interconnection method, which is based on finding Casimir functions for the closed-loop system.

### *5.2.1 Modeling and Control in a Hamiltonian Setting*

Let us introduce a state manifold $\mathcal{X}$ equipped with coordinates $(x^\alpha)$, $\alpha = 1, \ldots, n = \dim(\mathcal{X})$. A PCHD system, see [1, 14] can be formulated as

$$\begin{aligned} \dot{x} &= (J(x) - R(x))(\partial_x H)^T + G(x)u \\ y &= G^T(x)(\partial_x H)^T \end{aligned} \tag{5.1}$$

with the Hamiltonian[1] $H \in C^\infty(\mathcal{X})$. The maps $J$, $R$ and $G$ correspond to the interconnection $J$ (skew-symmetric), the damping $R$ (symmetric and positive semi-definite), and the input map $G$ (we will suppress the dependence on $x$ in $J$, $R$ and $G$ in the forthcoming). System inputs are denoted by $u$ and accordingly for the outputs we use $y$.

Let us consider a vector field $v$ on $\mathcal{X}$ (possibly depending on the input $u$), i.e.

$$v = \sum_{\alpha=1}^{n} v^\alpha(x,u)\partial_\alpha\,, \quad \partial_\alpha = \frac{\partial}{\partial x^\alpha}.$$

Then the change of $H$ in the direction of $v$ (Lie-derivative) reads as $v(H) = (\partial_x H)\, v$ and with $v = \dot{x}$ we obtain

$$v(H) = -(\partial_x H)R(\partial_x H)^T + y^T u \tag{5.2}$$

where we write $\dot{H} = v(H)$ in this special case. Obviously the relation (5.2) shows how the Hamiltonian is affected along solutions of the system, namely by dissipation and the collocation of the inputs and outputs. Additionally it is obvious that for $u = 0$ we have $\dot{H} \leq 0$ which gives the desired link to Lyapunov stability if $H$ is positive definite and serves as a Lyapunov function candidate for a desired equilibrium.

[1] Here $C^\infty(\mathcal{X})$ denotes the set of smooth functions on $\mathcal{X}$.

#### 5.2.1.1 Casimir Functions

A Casimir function for the system (5.1) is a function $C \in C^\infty(\mathcal{X})$ such that

$$\dot{C} = (\partial_x C)\dot{x} = (\partial_x C)Gu \tag{5.3}$$

holds independently of the Hamiltonian $H$, see [1]. This leads to the partial differential equation $\partial_x C(J-R) = 0$ in the unknown function $C$. If in particular $(\partial_x C)Gu = 0$ then $C$ is a constant of the motion since then $\dot{C} = 0$.

#### 5.2.1.2 Control by Interconnection

The focus of this part is to present the well-known control by interconnection method, which is based on the idea to couple PCHD systems via their energy ports. One system is to be controlled and the other one acts as the controller. Let us consider the system of the form

$$\begin{aligned} \dot{x} &= (J-R)(\partial_x H)^T + Gu \\ y &= G^T(\partial_x H)^T \end{aligned} \tag{5.4}$$

together with

$$\begin{aligned} \dot{x}_c &= (J_c - R_c)(\partial_{x_c} H_c)^T + G_c u_c \\ y_c &= G_c^T(\partial_{x_c} H_c)^T \end{aligned} \tag{5.5}$$

serving as the controller with the controller Hamiltonian $H_c$. The next step is to interconnect these two systems in a power preserving fashion, such that

$$y^T u + y_c^T u_c = 0 \tag{5.6}$$

is met. One choice to fulfill the relation (5.6) is a classical feedback of the form

$$u_c = Ky\,, \quad u = -K^T y_c. \tag{5.7}$$

Interconnecting (5.4) and (5.5) by means of (5.7) gives again a PCHD-System of the form

$$\dot{x}_{cl} = (J_{cl} - R_{cl})(\partial_{x_{cl}} H_{cl})^T$$

with $x_{cl} = (x, x_c)$ meeting

$$\dot{H}_{cl} = -(\partial_x H)R(\partial_x H)^T - (\partial_x H_c)R_c(\partial_{x_c} H_c)^T \leq 0$$

where the Hamiltonian of the closed loop reads as

$$H_{cl} = \underbrace{H(x)}_{Plant} + \underbrace{H_c(x_c)}_{Controller}.$$

Casimir functions of the closed-loop, which are conserved quantities, since $u_{cl} = 0$, take the form

$$C_{cl} = C_{cl}(x, x_c).$$

We are interested in $m$ functions of the special form

$$C^{\lambda}_{cl} = x^{\lambda}_c + C^{\lambda}(x)\,,\ \lambda = 1, \ldots, m$$

with $m \leq n$, since this allows for a relation of some of the plant and the controller states.

The Hamiltonian of the closed loop can be written as

$$H_{cl} = H(x) + H_c(x^{\lambda}_c, x^{\mu}_c)\,, \qquad \mu = m+1, \ldots, n_c$$

with $x_c = (x^{\lambda}_c, x^{\mu}_c)$, where this decomposition of the controller states is in connection with the ansatz for $C^{\lambda}_{cl}$ and since $\dot{C}_{cl} = 0$ we obtain with $x^{\lambda}_c + C^{\lambda}(x) = \kappa^{\lambda}$ the desired connection between some of the controller states and the plant states, where $\kappa$ follows from the initial conditions of the plant and the controller. This allows for a shaping of $H_{cl}$, such that possibly a desired equilibrium of the closed loop corresponds to the minimum of $H_{cl}$.

### 5.2.2 Example: The Pendulum with End-Mass

Let us consider a pendulum of length $l$ and mass $m$ whose suspension is allowed to move in horizontal direction (degree of freedom called $x$). The pendulum angle measured from the downward vertical position is denoted by $\phi$ and additionally an end-mass $m_e$ is attached. With $\dot{x} = v$ and $\dot{\phi} = \omega$ the kinetic energy is of the form

$$T = \frac{1}{2}(m + m_e)v^2 + \frac{1}{2}(m + 2m_e)l\omega v\cos(\phi) + \frac{1}{2}(m_e + \frac{m}{3})l^2\omega^2$$

and the potential energy reads as

$$V = -\Big(\frac{m}{2} + m_e\Big)gl(1 - \cos(\phi)).$$

The mass matrix can be deduced as

$$M = \begin{bmatrix} m + m_e & \left(\frac{m}{2} + m_e\right) l\cos(\phi) \\ \left(\frac{m}{2} + m_e\right) l\cos(\phi) & \left(m_e + \frac{m}{3}\right) l^2 \end{bmatrix}$$

and the Hamiltonian can be written as

$$H = \frac{1}{2} p^T M^{-1} p + V$$

with $p = [p_v \, p_\omega]^T = M \begin{bmatrix} v \\ \omega \end{bmatrix}$.

The corresponding Hamiltonian representation follows as

$$\begin{bmatrix} \dot{x} \\ \dot{\phi} \\ \dot{p}_v \\ \dot{p}_\omega \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \partial_x H \\ \partial_\phi H \\ \partial_{p_v} H \\ \partial_{p_\omega} H \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} F$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \partial_x H \\ \partial_\phi H \\ \partial_{p_v} H \\ \partial_{p_\omega} H \end{bmatrix},$$

where $F$ is the control input acting on the suspension point. For the controller system we choose a system of dimension 2 which is given as

$$\begin{bmatrix} \dot{q}_c \\ \dot{p}_c \end{bmatrix} = \left( \begin{bmatrix} 0 & J_{12} \\ -J_{12} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & r \end{bmatrix} \right) \begin{bmatrix} \partial_{q_c} H_c \\ \partial_{p_c} H_c \end{bmatrix} + \begin{bmatrix} G_{c1} \\ G_{c2} \end{bmatrix} u_c$$
$$y_c = G_{c1} \partial_{q_c} H_c + G_{c2} \partial_{p_c} H_c$$

with the controller Hamiltonian

$$H_c = \frac{1}{2} k_1 p_c^2 + \frac{1}{2} k_2 q_c^2, \quad k_1, k_2 > 0.$$

From the power conserving interconnection $F = -y_c$ and $u_c = y$, with respect to the simple choice $K = I$ (identity matrix $I$) we obtain the following closed-loop system

$$\begin{bmatrix} \dot{x} \\ \dot{\phi} \\ \dot{p}_v \\ \dot{p}_\omega \\ \dot{q}_c \\ \dot{p}_c \end{bmatrix} = \left( \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -G_{c1} & -G_{c2} \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & G_{c1} & 0 & 0 & J_{12} \\ 0 & 0 & G_{c2} & 0 & -J_{12} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r \end{bmatrix} \right) \times \begin{bmatrix} \partial_x H \\ \partial_\phi H \\ \partial_{p_v} H \\ \partial_{p_\omega} H \\ \partial_{q_c} H_c \\ \partial_{p_c} H_c \end{bmatrix}.$$

The choice $J_{12} = 0$ and $G_{c1} = G_{c2} = 1$ enables us to find the closed-loop Casimir function

$$C_{cl} = x - q_c$$

which has the consequence that for $x|_{t=0} = q_c|_{t=0}$ we have $x(t) = q_c(t)$ in the closed-loop due to $\dot{C}_{cl} = 0$. From the positive definite Hamiltonian

$$H_{cl} = \frac{1}{2} p^T M^{-1} p + V + \frac{1}{2} k_1 p_c^2 + \frac{1}{2} k_2 q_c^2$$

and

$$\dot{H}_{cl} = -r(k_c p_c)^2 \leq 0$$

we conclude that the equilibrium $q_c = x = \phi = p_c = \omega = v = 0$ is stable in the sense of Lyapunov. For the proof of asymptotic stability La-Salle's invariance principle can be employed [2], but since the counterpart in the PDE case is much more sophisticated, we also skip this part here.

## 5.3 The Infinite-Dimensional Case

The case of infinite-dimensional systems described by partial differential equations is much more involved compared to the finite-dimensional case. One of the quite obvious differences is the fact that we have to deal with Hamiltonian densities, e.g. quantities that can be integrated but it should be noted that they may depend on derivative variables in general. In this contribution we restrict ourselves to the case of mechanical systems with a one dimensional spatial domain, denoted by $\mathcal{D}$. Furthermore instead of a state manifold $\mathcal{X}$ we have to introduce a state bundle also

denoted by $\mathcal{X}$ fibred over $\mathcal{D}$ such that we have dependent coordinates $x$ and independent ones denoted by $X$. The state of the system is then a mapping $x = \Phi_t(X)$ for each $t$ corresponding to the time.

The Hamiltonian density reads as

$$\mathfrak{H} = \mathcal{H}(X,\ x,\ x_X)\mathrm{d}X$$

with a volume element $\mathrm{d}X$ on $\mathcal{D}$ where $x_X$ denotes the derivative variables, i.e. if $x = \Phi_t(X)$ is given then $x_X$ corresponds to $\partial_X \Phi_t(X)$. In the special case where $\mathfrak{H}$ corresponds to an energy density, then the total energy at an instant of time can be obtained by $\int_{\mathcal{D}} \mathcal{H}(X, x, x_X)\mathrm{d}X$ where a solution for $x = \Phi_t(X)$ has to be plugged in.

### 5.3.1 Modeling and Control: The Infinite Setting

A port controlled Hamiltonian system then takes the form of

$$\begin{aligned} \dot{x} &= (J - R)(\delta\mathfrak{H})^T + Gu \\ y &= G^T(\delta\mathfrak{H})^T \end{aligned} \tag{5.8}$$

and additional boundary conditions, see [10] and references therein.

**Remark 1.** *The differential geometric interpretation of the system* (5.8) *is quite sophisticated and additional explanation can be found* e.g. *in* [8, 10, 11]. *Furthermore the maps $J, R$ and $G$ can be differential operators as well in this scenario, but we will exclude this case here for simplicity.*

The Hamiltonian in this setting is a density $\mathfrak{H} = \mathcal{H}\mathrm{d}X$ as mentioned above and the variational derivative $\delta$ acts on $\mathcal{H}\mathrm{d}X$ in the following manner

$$(\delta\mathfrak{H})_\alpha = (\delta_\alpha \mathcal{H})\mathrm{d}X$$

with

$$\delta_\alpha \mathcal{H} = \partial_\alpha \mathcal{H} - d_X \partial_\alpha^X \mathcal{H}\,, \quad \partial_\alpha = \frac{\partial}{\partial x^\alpha}\,, \quad \partial_\alpha^X = \frac{\partial}{\partial x_X^\alpha}$$

where $d_X$ denotes the total derivative with respect to the independent variable $X$ and reads in our first order scenario as

$$d_X = \sum_{\alpha=1}^{n} (\partial_X + x_X^\alpha \partial_\alpha + x_{XX}^\alpha \partial_\alpha^X)$$

where $x_{XX}$ denote derivative variables of second order.

Let us consider in analogy to the finite-dimensional case a vector field which is used to measure the change of the Hamiltonian, i.e. the Hamiltonian density since we are dealing with field theories. We use a (generalized) vertical vector field $v$. Then we find[2]

$$\dot{\mathfrak{H}} = \int_{\mathcal{D}} (\delta_x \mathcal{H}) v \, \mathrm{d}X + \int_{\partial\mathcal{D}} \left(\partial_x^X \mathcal{H}\right) v \, \mathrm{d}X_B \tag{5.9}$$

where $v = \dot{x}$ and $\mathrm{d}X_B$ denotes the boundary (only two points $X = 0$ and $X = L$ in our setting, since $\mathcal{D}$ will be one dimensional). Here the notation

$$\delta_x \mathcal{H} = (\delta_{x^1} \mathcal{H}, \ldots, \delta_{x^n} \mathcal{H}), \quad \partial_x^X \mathcal{H} = \left(\frac{\partial \mathcal{H}}{\partial x_X^1}, \ldots, \frac{\partial \mathcal{H}}{\partial x_X^n}\right)$$

is used. Plugging in explicitly the PDEs (5.8), then (5.9) takes the form

$$\dot{\mathfrak{H}} = \int_{\mathcal{D}} -(\delta_x \mathcal{H}) \, R (\delta_x \mathcal{H})^T \, \mathrm{d}X + \int_{\mathcal{D}} y^T u \, \mathrm{d}X + \int_{\partial\mathcal{D}} \left(\partial_x^X \mathcal{H}\right) \dot{x} \, \mathrm{d}X_B$$

where again the dissipation and the collocation become apparent. In contrast to the finite-dimensional case the additional expression on the boundary appears, which in many applications forms a boundary port.

#### 5.3.1.1 Casimir Functionals

In the case of partial differential equations we consider Casimir densities (or functionals) and we restrict ourselves to the first order case only, i.e. $\mathfrak{C} = \mathcal{C}\mathrm{d}X$ such that $\mathcal{C}$ depends on first order derivatives. Using (5.9) where we replace $\mathcal{H}$ by $\mathcal{C}$ and setting $v = \dot{x}$ the relation (in analogy to (5.3))

$$\dot{\mathfrak{C}} = \int_{\mathcal{D}} (\delta_x \mathcal{C} \; G u) \mathrm{d}X$$

is obtained provided $\int_{\mathcal{D}} (\delta_x \mathcal{C})(J - R)(\delta_x \mathcal{H})^T \mathrm{d}X = 0$ is met and an additional boundary expression vanishes.

This leads to the following two conditions for the Casimir density

$$\delta_x \mathcal{C} \, (J - R) = 0 \tag{5.10}$$

$$\left(\partial_x^X \mathcal{C} \, \dot{x}\right)\big|_{\partial\mathcal{D}} = 0 \tag{5.11}$$

[2] Formally the prolongation of $v$ is used to derive $\dot{\mathfrak{H}}$ see [10] for details.

which have to be fulfilled. If in addition $\int_{\mathcal{D}} (\delta_x \mathcal{C} Gu) \mathrm{d}X = 0$ is met, then the density is again a conserved quantity.

#### 5.3.1.2 Control by Interconnection

In this section we discuss the interconnection (in a power conserving manner) of finite-dimensional Hamiltonian systems with an infinite-dimensional one, where we restrict ourselves to spatial domains which are one dimensional $X = [0, L]$ as well as to boundary control, see also [10, 11]. The boundary of the infinite-dimensional system is decomposed such that $\partial\mathcal{D} = \mathcal{D}_a \cup \mathcal{D}_u$ is met, where $\mathcal{D}_a$ denotes the actuated boundary ($X = L$) and $\mathcal{D}_u$ is the unactuated boundary ($X = 0$). The coupling will be performed by interconnecting the systems via energy ports. We will analyze the coupling of a finite-dimensional controller system with the infinite-dimensional system at $\mathcal{D}_a$. The infinite-dimensional system are partial differential equations in Hamiltonian representation modeled as

$$\dot{x} = (J - R)(\delta \mathfrak{H})^T \tag{5.12}$$

where the control enters through the actuated boundary $\mathcal{D}_a$. From (5.9) it becomes obvious that the energy port at the boundary (if it exists) can be expressed by

$$\partial_x^X \mathcal{H}\, \dot{x}\big|_{\partial\mathcal{D}} = y_\partial^T u_\partial. \tag{5.13}$$

Here $u_\partial$ and $y_\partial$ denote collocated inputs and outputs where the assignment of input or output is not unique, see [8] for more details. In the sequel we want to couple a finite-dimensional controller system at the actuated boundary (denoted by $a$).

The relation (5.13) for this configuration reads as

$$\partial_x^X \mathcal{H}\, \dot{x}\big|_{\mathcal{D}_a} = y_{\partial,a}^T u_{\partial,a}. \tag{5.14}$$

A power conserving interconnection has to fulfill the relation

$$y_c^T u_c + y_{\partial,a}^T u_{\partial,a} = 0 \tag{5.15}$$

where $y_c$ and $u_c$ denote the collocated port variables of the finite-dimensional controller system that reads as

$$\begin{aligned} \dot{x}_c &= (J_c - R_c)(\partial_{x_c} H_c)^T + G_c u_c \\ y_c &= G_c^T (\partial_{x_c} H_c)^T \end{aligned} \tag{5.16}$$

and is modeled on the manifold $\mathcal{X}_c$. The interconnection is chosen according to (5.15) as a feedback interconnection in the form

$$u_c = K\, y_{\partial,a}\,, \quad u_{\partial,a} = -K^T\, y_c \tag{5.17}$$

with an appropriate map $K$.

Now we are able to exploit the benefits of the control by interconnection technique as described for the ODE case in the last chapter. To this end we analyze Casimir functionals for the coupled systems in order to relate the states of the controller with the plant. We now have to find quantities of the form $C_I = C_c + \int_{\mathcal{D}} \mathcal{C}\mathrm{d}X$ with $C_c \in C^\infty(\mathcal{X}_c)$. The change of $C_I$ along solutions of the interconnected system (we assume again the existence of a solution) should vanish, i.e. $\dot{C}_I = 0$ since we are in a closed loop scenario and no further inputs are present.

The remarkable fact of this procedure and in special the choice for $C_I$ is the relation of (some of) the controller states with quantities of the plant to be controlled since $C_I$ is a constant along the solutions of the interconnected system, i.e. $x_c = -\int_{\mathcal{D}} \mathcal{C}\mathrm{d}X + \kappa$ where $\kappa$ depends on the initial conditions only. The Hamiltonian of the interconnected system $H_I = H_c + \int_{\mathcal{D}} \mathcal{H}\mathrm{d}X$ can be used for stability investigations provided that it serves as a Lyapunov function candidate, where in this context the connecting term $x_c = -\int_{\mathcal{D}} \mathcal{C}\mathrm{d}X + \kappa$ plays an extraordinary important role.

### 5.3.2 Example: The Heavy Chain System with End-Mass

In this section we demonstrate the proposed method using the heavy chain system with an end-mass $m_e$ exposed to gravity (acceleration due to gravity is denoted by $g$). We introduce the following bundle structure $\mathcal{X} \to \mathcal{D}$, $(X, w, p) \to X$, where $X$ is the coordinate of the one-dimensional spatial domain, $w$ denotes the deflection and $p$ the temporal momentum. The boundary $\partial\mathcal{D}$ consists of two points only, namely $X = 0$ and $X = L$ where $L$ is the length of the chain. Approximately, the system can be modeled by the partial differential equation $\rho\ddot{w} = d_X(P(X)w_X)$ which can also be stated as

$$\begin{aligned} \dot{w} &= \frac{p}{\rho} \\ \dot{p} &= d_X(P(X)w_X) \end{aligned} \tag{5.18}$$

where $\rho$ is the mass (line) density and the force in the chain reads as $P(X) = g\rho X + gm_e$. We consider the boundary conditions of the form

$$P(X)w_X|_{X=0} = m_e\, \ddot{w}|_{X=0}\,, \quad P(X)w_X|_{X=L} = F \tag{5.19}$$

and $F$ serves as the control input at the actuated boundary at $X = L$.

In a Hamiltonian setting we obtain with $x = (w, p)$ the desired representation

$$\begin{bmatrix} \dot{w} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \delta_w \mathcal{H} \\ \delta_p \mathcal{H} \end{bmatrix}$$

with

$$\delta_w \mathcal{H} = -d_X(P(X)w_X), \quad \delta_p \mathcal{H} = \frac{p}{\rho}$$

and the Hamiltonian density reads as

$$\mathcal{H} = \frac{1}{2\rho} p^2 + \frac{1}{2} P(X) w_X^2. \tag{5.20}$$

The total energy can be evaluated from

$$H = \int_0^L \left( \frac{1}{2\rho} p^2 + \frac{1}{2} P(X) w_X^2 \right) \mathrm{d}X + \frac{1}{2m_e} p_e^2$$

with $p_e = m_e \, \dot{w}|_{X=0}$.

Let us now investigate the control by interconnection problem where we use a finite-dimensional controller system of the form (5.16) where we choose $\dim(\mathcal{X}_c) = 2$ with $x_c = (q_c, p_c)$.

$$J_c = \begin{bmatrix} 0 & J_{12} \\ -J_{12} & 0 \end{bmatrix}, R_c = \begin{bmatrix} 0 & 0 \\ 0 & r \end{bmatrix}, r > 0, \quad G_c = \begin{bmatrix} G_{c,1} \\ G_{c,2} \end{bmatrix}$$

and the Hamiltonian of the finite-dimensional controller system can be chosen as

$$H_c = \frac{1}{2} k_1 p_c^2 + \frac{1}{2} k_2 q_c^2, \quad k_1, k_2 > 0.$$

The power conserving interconnection is given as

$$F = u_{\partial,a} = -y_c, \quad u_c = y_{\partial,a} = \dot{w}|_{X=L}$$

and consequently the coupled system read as

$$\begin{bmatrix} \dot{w} \\ \dot{p} \\ \dot{q}_c \\ \dot{p}_c \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{12} \\ 0 & 0 & -J_{12} & -r \end{bmatrix} \begin{bmatrix} \delta_w \mathcal{H} \\ \delta_p \mathcal{H} \\ \partial_{q_c} H_c \\ \partial_{p_c} H_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ G_{c,1} \\ G_{c,2} \end{bmatrix} \dot{w}|_{X=L}$$

with

$$P(X)w_X|_{X=0} = m_e\,\ddot{w}|_{X=0}\,,\;\; P(X)w_X|_{X=L} = -k_2(q_c - q_{c_d}) - k_1 p_c.$$

For $C_I$ we make the ansatz

$$C_I = q_c + \int_0^L \mathcal{C}\mathrm{d}X \tag{5.21}$$

and from $\dot{C}_I = 0$ we obtain

$$\dot{C}_I = \dot{q}_c + \int_0^L (\delta_x \mathcal{C})\,\dot{x}\mathrm{d}X + \left(\partial_x^X \mathcal{C}\right)\dot{x}|_0^L\,. \tag{5.22}$$

Choosing $J = 0$ and $G_{c,1} = G_{c,2} = 1$ then with

$$\mathcal{C} = -\frac{1}{L}d_X(Xw) = -\frac{1}{L}(w + Xw_X)$$

we see that (5.22) simplifies to

$$\dot{w}|_{X=L} - \frac{X}{L}\dot{w}\Big|_0^L = 0.$$

The Casimir $C_I$ for the closed loop is then obtained from (5.21) as

$$C_I = q_c - \frac{1}{L}Xw\Big|_0^L = q_c - w|_{X=L} = \kappa, \tag{5.23}$$

where $\kappa$ depends on the initial conditions only since $\dot{C}_I = 0$.

The Hamiltonian of the interconnected system can be written as

$$H_I = \int_0^L \left(\frac{1}{2\rho}p^2 + \frac{1}{2}P(X)w_X^2\right)\mathrm{d}X + \frac{1}{2m_e}p_e^2 + \frac{1}{2}k_1 p_c^2 + \frac{1}{2}k_2 q_c^2$$

and

$$\dot{H}_I = -rk_1^2 p_c^2 \leq 0.$$

This is formally the same result as in the finite-dimensional case. Unfortunately it is not possible to conclude that the desired equilibrium is stable – we will have to dig deeper using arguments from functional analysis, which will be presented in the forthcoming.

#### 5.3.2.1 Sketch of the Stability Proof

We start to formulate again the closed loop system in a manner appropriate for the following investigations, i.e. we have that

$$\begin{aligned}\dot{p}_m &= P(X)w_X|_{X=0}\\ \dot{p}_c &= -rk_1p_c + \frac{p}{\rho}\Big|_{X=L}\\ \dot{w} &= \frac{p}{\rho}\\ \dot{p} &= d_X(P(X)w_X)\end{aligned}$$

with the boundary conditions of the form

$$P(X)w_X|_{X=0} = m_e\,\ddot{w}|_{X=0},\;\; P(X)w_X|_{X=L} = -k_2(w|_{X=L}) - k_1p_c.$$

For the following analysis we have to define a function space, which will be denoted by $\mathcal{Z}$ meeting[3]

$$\mathcal{Z} = \{z = (p_m, p_c w, p) | p_m, p_c \in R,\, w \in H^1, p \in L^2\}$$

which is a Hilbert space, however equipped with the (equivalent) inner product of the form

$$\langle z, \bar{z}\rangle_{H_I} = \int_0^L \Big(\frac{1}{\rho}p\bar{p} + P(X)w_X\,\bar{w}_X\Big)\mathrm{d}X + \frac{1}{m_e}p_m\,\bar{p}_m + k_1p_c\,\bar{p}_c + k_2w(L)\bar{w}(L)$$

which is connected to the closed loop Hamiltonian $H_I$.

**Remark 2.** *In $\mathcal{Z}$ the natural inner product is given as*

$$\langle z, \bar{z}\rangle_n = \frac{1}{2}p_m\,\bar{p}_m + \frac{1}{2}p_c\,\bar{p}_c + \frac{1}{2}\int_0^L (w\bar{w} + w_X\,\bar{w}_X)\mathrm{d}X + \frac{1}{2}\int_0^L p\bar{p}\;\mathrm{d}X$$

*inducing the natural norm $\|z\|_n^2 = \langle z, z\rangle_n$. Since we are using $\langle z, \bar{z}\rangle_{H_I}$ we have to fulfill*

$$C_1\,\|z\|_n^2 \leqq \|z\|_{H_I}^2 \leqq C_2\,\|z\|_n^2\;, \quad C_1, C_2 > 0$$

[3] Here $H^k$ with $k = 0, 1, 2, \ldots$ denote the Sobolev spaces over the domain $(0, L)$ which roughly speaking consists of functions whose derivatives up to order $k$ are quadratic integrable in the sense of Lebesque, i.e. $L^2$. In particular we have $H^0 = L^2$.

*in order to show the equivalence of the norms with* $\|z\|_{H_I}^2 = \langle z, z\rangle_{H_I}$. *Meeting the lower bound can be shown easily using the following inequality*

$$C_3 \int_0^L w^2 \mathrm{d}X \leqq w^2(L)\,, \quad C_3 \geq 0$$

*whereas for the upper bound the Sobolev embedding Theorem has to be applied, see also* [15] *and references therein.*

Next we formulate the so-called Cauchy problem with $\dot{z} = \mathcal{A}z$ reading as

$$\begin{bmatrix} \dot{p}_m \\ \dot{p}_c \\ \dot{w} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} P(X)w_X|_{X=0} \\ -rk_1p_c + \frac{p(L)}{\rho} \\ \frac{p(X)}{\rho} \\ \partial_X(P(X)\partial_X w(X)) \end{bmatrix}$$

with

$$D(\mathcal{A}) = \{p_m, p_c, w, p|, p_m, p_{,c} \in \mathbb{R}, w \in H^2, p \in H^1,\, p_m = m_e \frac{p}{\rho}\Big|_{X=0},$$
$$P(X)w_X|_{X=L} = -k_2(w|_{X=L}) - k_1p_c\}.$$

The proof of stability is based on the following scheme: If $\mathcal{A}$ generates a contracting $C_0$ semi-group $T(t)$ then from $z(t) = T(t)z_0$ and

$$\|T(t)\| = \sup_{z_0 \in \mathcal{Z}\setminus 0} \frac{\|T(t)z_0\|_{H_I}}{\|z_0\|_{H_I}} \leq 1$$

we obtain finally

$$\|z(t)\|_{H_I} \leq \|T(t)\|_{H_I}\, \|z_0\|_{H_I} \leq \|z_0\|_{H_I}$$

from which we conclude stability of the desired equilibrium in the sense of Lyapunov with regard to $\|\cdot\|_{H_I}$. To show that $\mathcal{A}$ in fact generates a contracting $C_0$ semi-group we will apply a variant of the so-called Lümer Phillips Theorem, see [12, 13, 16, 17], which means in our concrete setting, that $\mathcal{A}$ has to be a dissipative operator, and its inverse has to exist and must be bounded. $\mathcal{A}$ is dissipative, since $\langle z, \mathcal{A}z\rangle_{H_I} \leq 0$ follows by $\dot{H}_I = -rk_1^2p_c^2 \leq 0$ and concerning $\mathcal{A}^{-1}$ we construct the inverse of $\bar{z} = \mathcal{A}z$ by applying partly the integrating by parts technique. In fact, from

$$\begin{bmatrix} \bar{p}_m \\ \bar{p}_c \\ \bar{w} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} P(X)w_X|_{X=0} \\ -rk_1p_c + \frac{p(L)}{\rho} \\ \frac{p(X)}{\rho} \\ \partial_X(P(X)\partial_X w(X)) \end{bmatrix}$$

we deduce $p = \rho\bar{w}$ and $p_c = -\frac{1}{rk_1}(\bar{p}_c - \bar{w}(L))$. Furthermore from

$$\begin{aligned} \int_0^X \bar{p}(Z^1)\mathrm{d}Z^1 &= P(X)\partial_X w(X) - P(0)\partial_X w(0) \\ &= P(X)\partial_X w(X) - \bar{p}_m \end{aligned}$$

we easily derive

$$\int_X^L \frac{1}{P(Z^2)} \int_0^{Z^2} \bar{p}(Z^1)\mathrm{d}Z^1\mathrm{d}Z^2 = w(L) - w(X) - \int_X^L \frac{\bar{p}_m}{P(Z^2)}\mathrm{d}Z^2.$$

In order to obtain $w(X)$ we have to determine $w(L)$ which can be done by evaluating

$$\int_0^L \bar{p}(Z^1)\mathrm{d}Z^1 + \bar{p}_m = P(L)\partial_X w(L) = -k_2 w(L) - k_1 p_c$$

leading to

$$w(L) = -\frac{1}{k_2}\int_0^L \bar{p}(Z^1)\mathrm{d}Z^1 + \left(\frac{1}{rk_2}(\bar{p}_c - \bar{w}(L))\right) - \frac{\bar{p}_m}{k_2}.$$

Finally we find

$$\begin{aligned} w(X) = &-\frac{1}{k_2}\int_0^L \bar{p}(Z^1)\mathrm{d}Z^1 + \left(\frac{1}{rk_2}(\bar{p}_c - \bar{w}(L))\right) - \frac{\bar{p}_m}{k_2} \\ &- \int_X^L \frac{1}{P(Z^2)} \int_0^{Z^2} \bar{p}(Z^1)\mathrm{d}Z^1\mathrm{d}Z^2 - \int_X^L \frac{\bar{p}_m}{P(Z^2)}\mathrm{d}Z^2 \end{aligned}$$

which guarantees that $\mathcal{A}^{-1}$ exists and is bounded for bounded $\bar{z}$ and $z = \mathcal{A}^{-1}\bar{z}$.

**Remark 3.** *The analysis of the asymptotic stability in the infinite-dimensional setting can be performed again by using La Salle's principle, however compared to the ODE scenario the approach is much more involved in the PDE case, see* [13, 17] *and references therein.*

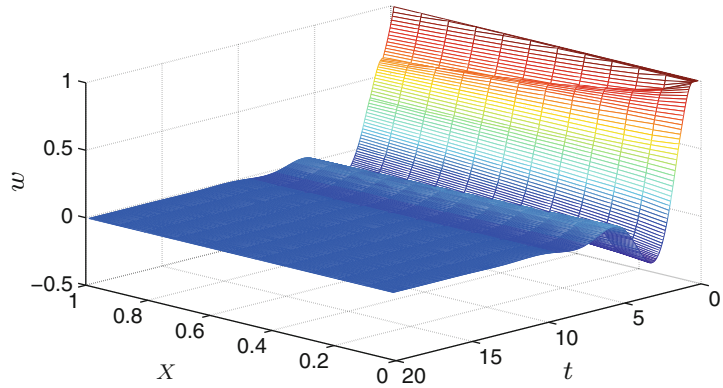



**Fig. 5.1** Simulation results for the deflection $w$

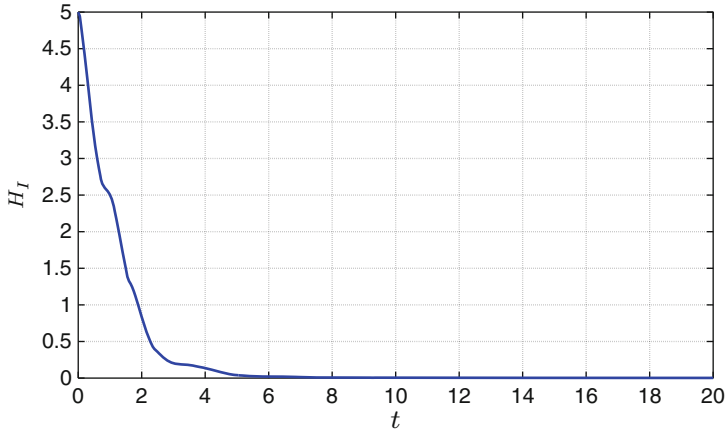



**Fig. 5.2** Simulation results for the closed loop Hamiltonian $H_I$

### 5.3.3 Simulation Result

Finally in Figs. 5.1 and 5.2 we present a simulation result of the controlled heavy chain system. The control objective is to stabilize the chain around a desired equilibrium (set-point) where the initial condition does not correspond to the desired set-point equilibrium.

We consider the simple but demonstrative case where the physical parameters

$$L = 1, \rho = 2.7, g = 10, m_e = 1$$

and the parameters of the controller are chosen as $k_1 = 400$, $k_2 = 10$, $r = 0.1$. Furthermore $w_d|_{X=L} = 0$. The initial conditions for the plant were chosen as

$w|_{t=0} = 1$, $p|_{t=0} = 0$, and for the controller $q_c|_{t=0} = 1, p_c|_{t=0} = 0$ such that $\kappa = 0$ follows immediately.

## 5.4 Conclusion

In this paper the port-Hamiltonian concept has been used with respect to the desire to obtain a modelling which stresses the physics behind the system equations together with the focus on energy based control concepts leading to stable closed-loop systems. This has been applied to systems described by ODEs first and then in a second step the concepts have been adapted to problems described by partial differential equations. The most interesting fact is that from the perspective of the design process the finite- and the infinite-dimensional case are very similar, which is no longer true when the proof of stability is the task.

**Acknowledgement** M. Schöberl is an APART fellowship holder of the Austrian Academy of Sciences.

## References

1. van der Schaft A (2000) L2-gain and passivity techniques in nonlinear control. Springer, New York
2. Khalil H (1996) Nonlinear systems. Prentice Hall Inc., New Jersey
3. Macchelli A, van der Schaft A, Melchiorri C (2004) Port Hamiltonian formulation of infinite dimensional systems: Part I. Modeling. In: Proceedings of the 43rd IEEE conference decision and control (CDC), Atlantis, Paradise Island, Bahamas, pp 3762–3767
4. Macchelli A, van der Schaft A, Melchiorri C (2004) Port Hamiltonian formulation of infinite dimensional systems: Part II. Boundary control by interconnection. In: Proceedings of the 43rd IEEE conference decision and control (CDC), Atlantis, Paradise Island, Bahamas, pp 3768–3773
5. Maschke BM, van der Schaft A (2005) Compositional modelling of distributed-parameter systems. Advanced topics in control systems theory. Springer lecture notes in control and information sciences, Springer-Verlag London Ltd, London
6. van der Schaft A, Maschke B (2002) Hamiltonian formulation of distributed-parameter systems with boundary energy flow. J Geom Phys 42:166–194
7. Schlacher K (2008) Mathematical modeling for nonlinear control: a Hamiltonian approach. Math Comput Simul 97:829–849
8. Schöberl M, Ennsbrunner H, Schlacher K (2008) Modelling of piezoelectric structures – a Hamiltonian approach. Math Comput Model Dyn Syst 14:179–193
9. Olver P (1986) Applications of Lie groups to differential equations. Springer, New York
10. Schöberl M, Siuka A (2011) On Casimir functionals for field theories in port-Hamiltonian description for control purposes. In: Proceedings 50th IEEE conference on decision and control (CDC), Orlando, Florida, pp 7759–7764
11. Siuka A, Schöberl M, Schlacher K (2011) Port-Hamiltonian modelling and energy-based control of the Timoshenko beam – an approach based on structural invariants. Acta Mech 222:69–89

12. Liu Z, Zheng S (1999) Semigroups associated with dissipative systems. Chapman & Hall/CRC, Boca Raton
13. Luo ZH, Guo BZ, Morgül O (1999) Stability and stabilization of infinite dimensional systems with applications. Springer, London
14. Ortega R, van der Schaft A, Maschke B, Escobar G (2002) Interconnection and damping assignment passivity-based control of port Hamiltonian systems. Automatica 36:585–596
15. Curtain R, Zwart H (1995) An introduction to infinite dimensional linear systems theory. Springer, New York
16. Morgül O (1998) Stabilization and disturbance rejection for the wave equation. IEEE Trans Autom Control 43(1):89–95
17. Thull D (2010) Tracking control of mechanical distributed parameter systems with applications. Shaker Verlag, Germany

# Chapter 6
# Passivity-Based Tracking Control of a Flexible Link Robot

**Peter Staufer and Hubert Gattringer**

**Abstract** This contribution addresses modeling and control of highly complex nonlinear mechanical systems such as an articulated robot with two flexible links and three flexible joints. We employ the Projection Equation in subsystem formulation, a very efficient method for modeling repeating assemblies and beam elasticities and apply a Ritz expansion to obtain ordinary differential equations of motion. For model-based control design, the small elastic deformations of the beams are approximated with linear springs and dampers in a *lumped element model*. On this basis, a control design with two degrees of freedom is proposed: a flatness-based feed forward and a passivity-based feedback control technique of interconnection and damping assignment. Further, we deal with acceleration and angular rate measurements to compute all system states used in the feedback loop. Finally, the proposed strategies are validated by measurements from a fast straight line in space and a ball catching scenario.

## 6.1 Introduction

Lightweight structures may offer a good trade-off between high acceleration (and thus short cycle times) on the one hand and good tracking accuracy on the other. However, reducing inertia parameters and the weight of robots may lead to a loss in stiffness, resulting in elastic deformations and vibrations at the tool center point (TCP). Therefore, rational model-based control design must avoid or compensate for this weakness.

In industrial setups, effective vibration suppression requires further measurements, in addition to motor position detection at each servo drive. For this purpose, we propose using acceleration and angular rate sensors, as they are inexpensive and easy to mount.

P. Staufer (✉) • H. Gattringer
Institute for Robotics, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria
e-mail: peter.staufer@jku.at; hubert.gattringer@jku.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_6, 

### 6.1.1 Modeling

A very comprehensive survey of the most commonly used modeling techniques and approaches for flexible robotic manipulators can be found in Ref. [1]. The Langrange formalism of the second kind and the Hamilton representation are well-established methods (Refs. [3, 4] provide excellent overviews). In contrast the Projection Equation is a very effective procedure by means of modeling the complex dynamics of highly nonlinear systems [2]. The Projection Equation is usually used for mechanical systems consisting of a large number of bodies with many degrees of freedom. The elastic links of the robot are assumed to be Euler-Bernoulli beams, and elasticities are modeled with Ritz Ansatz functions.

Conventionally, dealing with energy-based control concepts requires analytic procedures using energy functionals. As mentioned above, a reduced model in which the elasticities are approximated with lumped elements (LEM) is introduced, also called *mass-spring-damper models* or *virtual spring-damper-elements*, see Refs. [5] or [6].

### 6.1.2 Tracking Control

Flexible manipulator control has been the focus of extensive research (e.g., see Refs. [7, 8]). Some practical examples can be found in Refs. [9, 10], where linear robots with flexible beams were considered. In Ref. [11], robot similar to that considered in our work was discussed, although we use the flatness-based approach to solve the tracking problem, see Ref. [12]. In Ref. [13], a general algorithm for the inverse dynamics computation of robots with elastic links/joints was presented, but there are some uncertainties in the elastic system – which are discussed below – that prevent an exactly linearizing tracking control from achieving the desired results. In the present work, we therefore propose a *control structure with two degrees of freedom* (2DoF) for solving the tracking problem. Such an approach allows designing the feedforward part independently of the feedback part. The feedforward control is based on the flatness approach, while the feedback control of the remaining tracking error dynamics builds upon energy-based concepts. Many different approaches regarding this technique can be found in the literature. A general overview of passivity-based control can be found in Refs. [14, 15]. The fundamental idea of these approaches is energy preservation. Roughly speaking, energy ports of external inputs and collocated outputs are considered. These ports describe the power exchange between the system environment and the system dynamics. On this basis, we propose the *interconnection and damping assignment – passivity based control* (IDA-PBC) approach.

## 6.2 Setup

Figure 6.1 shows the mechanical setup under consideration. Fast trajectories and the influence of gravity result in two bending and torsional deflections of both arms. Triple-axis analog angular rate and acceleration sensors are mounted on the elbow and on the TCP to improve accuracy and minimize vibration. This sensor pack is called IMU (*internal measurement unit*) – a low-cost product that operates on a *micro-electro-mechanical system* (MEMS) principle. The three synchronous motors of the robot are powered by servo drives with a common cascaded motor joint control. Communication with a central computing unit and the servo drives is realized via a very fast Ethernet Powerlink bus.

## 6.3 System Dynamics

As mentioned in the introduction, various modeling procedures are used in this contribution. The Projection Equation and the canonical Hamilton Equations serve as the fundamental bases for a Ritz Model and a LEM.

### *6.3.1 Methods*

The equations of motion in configuration space can be written as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}, \tag{6.1}$$

with the mass matrix $\mathbf{M}$, minimal coordinates $\mathbf{q}$ and their time derivatives, the vector of nonlinearities $\mathbf{g}$, and the generalized force vector $\mathbf{Q}$. In state space, this reads

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{6.2}$$

with the state vector $\mathbf{x}^T = (\mathbf{q}^T \ \dot{\mathbf{q}}^T)$, the output vector $\mathbf{y}$, and the input vector $\mathbf{u}$.

#### 6.3.1.1 Projection Equation

The Projection Equation for $N_B$ bodies is given by

$$\sum_{i=1}^{N_B} \left[ \left( \frac{\partial {}_R\mathbf{v}_c}{\partial \dot{\mathbf{q}}} \right)^T \left( \frac{\partial {}_R\boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \right)^T \right]_i \left[ \begin{matrix} {}_R\dot{\mathbf{p}} + {}_R\tilde{\boldsymbol{\omega}}_{IR}\, {}_R\mathbf{p} - {}_R\mathbf{f}^e \\ {}_R\dot{\mathbf{L}} + {}_R\tilde{\boldsymbol{\omega}}_{IR}\, {}_R\mathbf{L} - {}_R\mathbf{M}^e \end{matrix} \right]_i = 0, \tag{6.3}$$

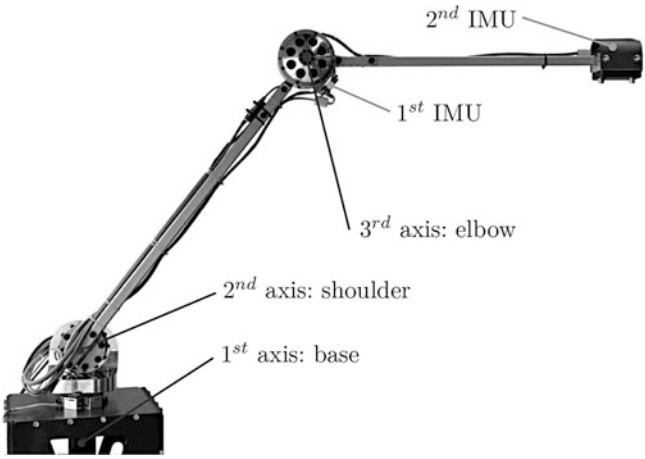



**Fig. 6.1** The elastic articulated robot is driven by three synchronous motors and *Harmonic Drive* gears for base, shoulder and elbow. The lightweight upper arm and forearm have square hollow cross-sections

see Ref. [2] for details. This method projects the linear and angular momenta ${}_R\mathbf{p}$ and ${}_R\mathbf{L}$ respectively, the impressed forces ${}_R\mathbf{f}^e$ and impressed moments ${}_R\mathbf{M}^e$, formulated in an arbitrary reference frame $R$, using the Jacobian matrices $(\partial_R \mathbf{v}_c/\partial\dot{\mathbf{q}})_i$ and $(\partial_R \boldsymbol{\omega}_c/\partial\dot{\mathbf{q}})_i$ (in terms of the selected minimal velocities $\dot{\mathbf{q}}$) into the direction of unconstrained motion. Variables ${}_R\mathbf{v}_c$ and ${}_R\boldsymbol{\omega}_c$ denote respectively the linear and angular velocities of the center of gravity for each body $i$. For repeating system modules as in our case (the upper arm and forearm, for instance), it is convenient to use the subsystem formulation of the Projection Equation:

$$\sum_{n=1}^{N} \left( \frac{\partial \dot{\mathbf{y}}_n}{\partial \dot{\mathbf{q}}} \right)^T \left\{ \mathbf{M}_n \ddot{\mathbf{y}}_n + \mathbf{G}_n \dot{\mathbf{y}}_n - \mathbf{Q}_n^e \right\} = 0, \tag{6.4}$$

combining similar segments of the rigid-elastic multibody system in a modular way. Each subsystem $n$ is associated with describing velocities $\dot{\mathbf{y}}_n$, a mass matrix $\mathbf{M}_n$, a matrix of Coriolis and centrifugal terms $\mathbf{G}_n$, and generalized forces $\mathbf{Q}_n^e$. For elastic multibody systems, however, $\dot{\mathbf{y}}_n$ in the Projection Equation (6.4) contain spatial derivatives of the holonomic minimal velocities $\dot{\mathbf{q}}$, due to the dependencies on space and time of the bending functions and torsion. A Ritz approximation for the distributed parameters is applied to obtain ordinary differential equations; see Ref. [2] for details.

#### 6.3.1.2 Canonical Hamiltonian Equations

Using the canonical Hamilton Equations requires the Legendre transformation

$$\mathbf{p} = \left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right)^T = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}, \tag{6.5}$$

where $T$ denotes the kinetic energy and $\mathbf{p}$ the generalized momenta. Therefore, $\dot{\mathbf{q}}$ must be replaced with $\dot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\mathbf{p}$ in the Hamiltonian function, which leads to

$$H(\mathbf{q}, \mathbf{p}) = T(\mathbf{q}, \mathbf{p}) + V(\mathbf{q}), \tag{6.6}$$

with the potential energy $V$. We thus yield for the Hamilton Equations

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & \mathbf{E} \\ -\mathbf{E} & -\mathbf{R}_m \end{bmatrix} \begin{pmatrix} \left(\frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{q}}\right)^T \\ \left(\frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{p}}\right)^T \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{Q} \end{pmatrix}, \tag{6.7}$$

with $\mathbf{x}^T = (\mathbf{q}^T\ \mathbf{p}^T)$. If the system representation with an affine system input $\mathbf{u}$ can be given in

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})) \left(\frac{\partial H}{\partial \mathbf{x}}\right)^T + \mathbf{G}(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{G}(\mathbf{x})^T \left(\frac{\partial H}{\partial \mathbf{x}}\right)^T, \end{aligned} \tag{6.8}$$

the system is called a *port-controlled Hamiltonian system with dissipation* (PCHD). In this context, $\mathbf{Q} = \mathbf{B}(\mathbf{x})\mathbf{u}$ with $\mathbf{G}(\mathbf{x})^T = (0\,\mathbf{B}(\mathbf{x})^T)$ is applied. The output vector $\mathbf{y}$ is defined as a collocated output to the input vector $\mathbf{u}$. The product of these variables $\mathbf{y}^T\,\mathbf{u}$ characterizes an energy port and represents the power flow into the system from the environment, while $\mathbf{J}(\mathbf{x}) = -\mathbf{J}(\mathbf{x})^T$ represents the internal power flow and $\mathbf{R}(\mathbf{x}) = \mathbf{R}(\mathbf{x})^T$ the dissipative effects in the system. More details can be found in Ref. [16].

### *6.3.2 Ritz Model*

Here we present how – building on the foundations outlined in Sect. 6.3.1.1 – the equations of motion can be computed by decomposing the system into subsystems. The $n^{\text{th}}$-subsystem of the Ritz model includes a motor-gear device and an elastic beam as is the case for the upper arm and the forearm pictured in Fig. 6.2. The vector of describing velocities of such a subsystem,

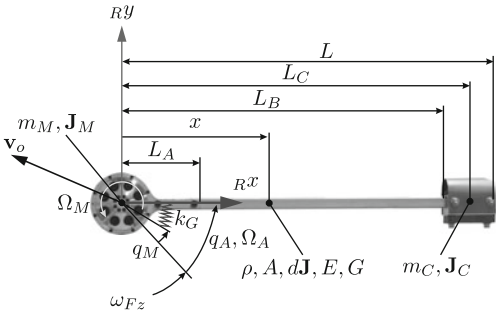



**Fig. 6.2** An elastic subsystem which includes a motor-gear device and an elastic beam such as the upper arm and the forearm

$$\dot{\mathbf{y}}_n^T = \left(\mathbf{v}_o^T \ \boldsymbol{\omega}_F^T \ \Omega_M \ \Omega_A \ \dot{\mathbf{q}}_R^T\right)_n, \tag{6.9}$$

contains the guiding velocities $\mathbf{v}_o, \boldsymbol{\omega}_F$ and the relative angular velocities $\Omega_A$, $\Omega_M$. The latter represent the arm and motor rotations relative to the moving frame. In this context for an infinitesimal element $dm$, the vector to the center of gravity is written as $\mathbf{r}_c^T = (x \ v(x,t) \ w(x,t))$ and the vector of rotation angles as $\boldsymbol{\varphi}^T = (\vartheta(x,t) - w'(x,t) \ v'(x,t))$. For an approximation of the elastic deflection $v(x,t) = \mathbf{v}(x)^T \mathbf{q}_v$ in the ${}_R y$-direction and $w(x,t) = \mathbf{w}(x)^T \mathbf{q}_w$ in the ${}_R z$-direction as well as $\vartheta(x,t) = \boldsymbol{\vartheta}(x)^T \mathbf{q}_\vartheta$ for torsion in the ${}_R x$-direction, Ritz Ansatz functions are introduced. Therefore, $\dot{\mathbf{y}}_n$ also contains the time derivatives of the Ritz coordinates $\mathbf{q}_R^T = (\mathbf{q}_v^T \ \mathbf{q}_w^T \ \mathbf{q}_\vartheta^T)$. All shape functions as well as the infinitesimal mass element $dm$ and the infinitesimal moments of inertia $d\mathbf{J}$ are defined piecewise in order to include the rigid parts ($x < L_A$ and $x > L_B$) of the link. The gear torque depends on the difference between arm and motor angles $q_A - q_M$ with the gear elasticity $k_G$. Finally, the motor unit is considered with $m_M$ and $\mathbf{J}_M$. The angular velocities are defined as $\Omega_A = \dot{q}_A$ and $\Omega_M = i_G \, \dot{q}_M$ with the gear ratio $i_G$. The resulting ordinary differential equations of motion are given in form of (6.1) with $\mathbf{q}^T = (\mathbf{q}_M^T \ \mathbf{q}_A^T \ \mathbf{q}_{el}^T)$, $\mathbf{q}_M^T \in \mathbb{R}^3$, $\mathbf{q}_A^T \in \mathbb{R}^3$ and $\mathbf{q}_{el}^T \in \mathbb{R}^{10}$, where five Ritz coordinates are included for each arm. More information on the modeling of elastic robots can be found in Refs. [17, 18].

### 6.3.3 Lumped Element Model

Alternatively, we propose a relatively simple model – based on lumped elements – for control design. This technique is justified since the considered elastic robot fulfills the assumption of small elastic deformations. Various investigations have shown that only three springs are essential for good concordance with the real model: $k_1$, $k_2$, and $k_3$ for elasticity in the base, the shoulder, and in the elbow respectively (see Fig. 6.3). Obviously, all torsional effects and the elasticity in the

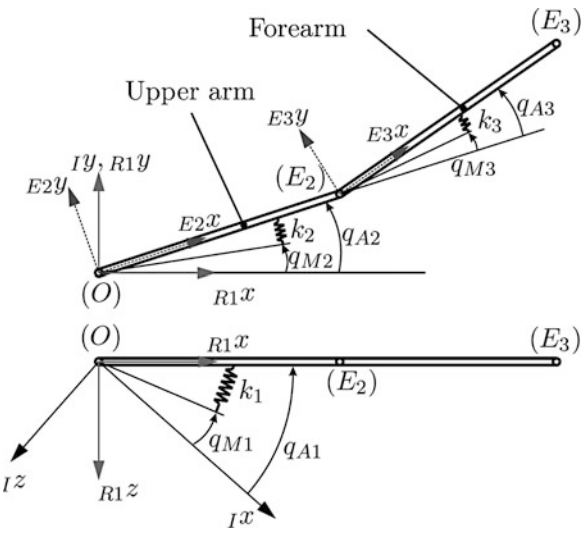



**Fig. 6.3** *Lumped Element Model* limited to the first deformation mode

$z$-direction of the forearm are neglected in this model. In order to reintroduce these effects, a position-dependent spring stiffness for the base $k_1(q_{A2}, q_{A3})$ (determined experimentally) is suggested. The equations of motion can be decomposed into a system of differential equations for motor movements (subscript $M$) and a system of arm equations (subscript $A$)

$$\mathbf{Q}_M = \mathbf{M}_M\,\ddot{\mathbf{q}}_M + \mathbf{Q}_A + \mathbf{Q}_F \tag{6.10}$$

$$\mathbf{Q}_A = \mathbf{M}_A\,\ddot{\mathbf{q}}_A + \mathbf{G}_A\,\dot{\mathbf{q}}_A + \mathbf{Q}_{G,A} \tag{6.11}$$

$$\mathbf{Q}_A = \mathbf{K}(\mathbf{q}_M - \mathbf{q}_A), \tag{6.12}$$

with a motor torque vector $\mathbf{Q}_M$, a friction vector $\mathbf{Q}_F$, and a vector with gravitational dependences $\mathbf{Q}_{G,A}$. The algebraic (6.12) links (6.10) to (6.11), where $\mathbf{K}$ denotes the stiffness matrix. The equations listed above constitute the fundamental basis for the following control design.

### *6.3.4 Model Verification*

Clearly, mathematical description and experiment must coincide for model-based control design. In order to obtain a meaningful comparison, the desired reference trajectory is selected in the following manner: In the first part of verification, each axis is moved separately. In the second part, a straight line in space with a length

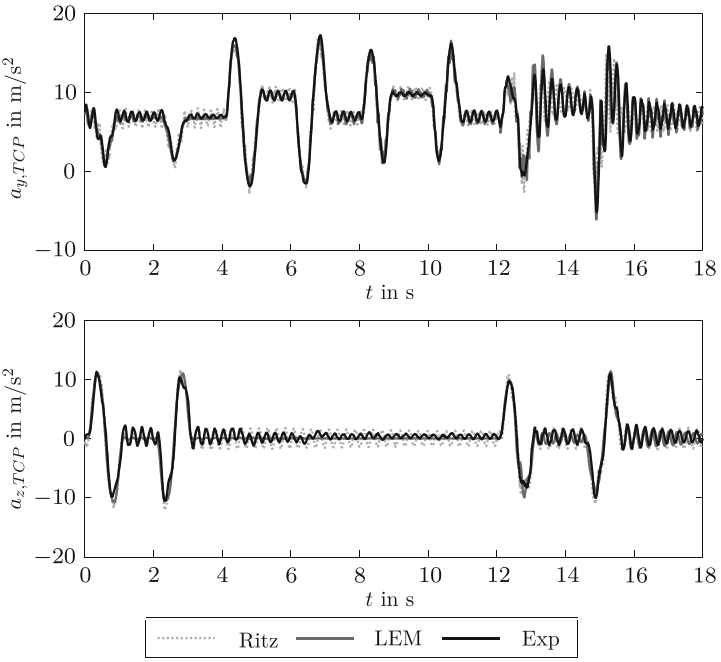



**Fig. 6.4** Simulated and measured TCP accelerations

of 2.1 m and a duration of 1.2 s is tracked. For a comparative study in terms of dynamic model matching, the same cascaded PD motor joint parameters are used in simulation and experiment. The plots in Fig. 6.4 show that simulation and experimental results for both the Ritz model and the LEM are strongly concordant. Since the Ritz model typically includes numerous parameters, the model is very close to the experiment, but also sensitive to parameter uncertainties. The LEM, in contrast, is relatively straightforward, as it depends only on a few parameters; hence, it is suitable for control design.

## 6.4 Computation of the Arm Positions and Velocities

If the LEM is used for control design, the feedback loop requires knowledge of the state vector

$$\mathbf{x}^T = (\mathbf{q}_M^T \ \mathbf{q}_A^T \ \dot{\mathbf{q}}_M^T \ \dot{\mathbf{q}}_A^T). \tag{6.13}$$

Motor position $\mathbf{q}_M$ and motor velocity $\dot{\mathbf{q}}_M$ are available on each servo drive. Next we show a straightforward calculation of the unknown arm position $\mathbf{q}_A$ and arm velocity $\dot{\mathbf{q}}_A$ with the installed IMUs. The main idea is a rebuild of the equations of motion with the acceleration sensors

$$ {}_i\mathbf{a}_{IMU,i-1} = (a_{x,i}\, a_{y,i}\, a_{z,i})^T = {}_i\dot{\mathbf{v}}_i + {}_i\tilde{\boldsymbol{\omega}}_{Ii}\ {}_i\mathbf{v}_i - {}_i\mathbf{g}_i \tag{6.14} $$

and the angular rate sensors

$$ {}_i\boldsymbol{\omega}_{IMU,i-1} = (\omega_{x,i}\, \omega_{y,i}\, \omega_{z,i})^T = {}_i\boldsymbol{\omega}_{Ii} \tag{6.15} $$

with $i = 2$ for the upper arm frame and $i = 3$ for the forearm frame (Fig. 6.3). In the first step, the mathematical model with $N_B$ bodies is divided into $N_1$ driving units and $N_2$ arm units. In the considered configuration, the interconnections between these units equate to spring forces with

$$ \mathbf{Q}_A = \mathbf{K}(\mathbf{q}_M - \mathbf{q}_A). \tag{6.16} $$

These findings can easily be taken into account in the Projection Equation (6.3), by replacing the projection in the direction of unconstrained motion with the projection in arm-space according to

$$ \sum_{i=1}^{N_2} \left[ \left( \frac{\partial_i \mathbf{v}_c}{\partial\, \dot{\mathbf{q}}_A} \right)^T \left( \frac{\partial_i \boldsymbol{\omega}_c}{\partial\, \dot{\mathbf{q}}_A} \right)^T \right]_i \left[ \begin{matrix} {}_i\dot{\mathbf{p}} + {}_i\tilde{\boldsymbol{\omega}}_{Ii}\, {}_i\mathbf{p} - {}_i\mathbf{f}^e \\ {}_i\dot{\mathbf{L}} + {}_i\tilde{\boldsymbol{\omega}}_{Ii}\, {}_i\mathbf{L} - {}_i\mathbf{M}^e \end{matrix} \right]_i = \mathbf{Q}_A, \tag{6.17} $$

where $\mathbf{Q}_A$ appears on the right-hand side. If we treat the beam and the tip body as one unit, there are two arm units (upper arm unit and forearm unit), and therefore $N_2 = 2$. If the IMUs are mounted at $l_i$, we yield with (6.14) and (6.15) and ${}_i\mathbf{p}_i = m_i\, {}_i\mathbf{v}_i$, ${}_i\mathbf{f}_i^e = m_i\, {}_i\mathbf{g}_i$, ${}_i\mathbf{L}_i = {}_i\mathbf{J}_i\, {}_i\boldsymbol{\omega}_{Ii}$, $\mathbf{M}^e = 0$:

$$ \mathbf{Q}_A = \sum_{i=2}^{3} \left[ \left( \frac{\partial_i \mathbf{v}_c}{\partial\, \dot{\mathbf{q}}_A} \right)^T \left( \frac{\partial_i \boldsymbol{\omega}_c}{\partial\, \dot{\mathbf{q}}_A} \right)^T \right]_i \cdot \left[ \begin{matrix} m_i\, {}_i\mathbf{a}_{IMU,i-1} \\ {}_i\mathbf{J}_i\, {}_i\dot{\boldsymbol{\omega}}_{IMU,i-1} + {}_i\tilde{\boldsymbol{\omega}}_{IMU,i-1}\ {}_i\mathbf{J}_i\, {}_i\boldsymbol{\omega}_{IMU,i-1} \end{matrix} \right]. \tag{6.18} $$

Taking (6.16) into account, the arm position can be computed solving (6.18) recursively, which is shown in the next steps. However, since the considered robot features two heavy tip bodies on each link, all arm inertia can be neglected, because the Steiner term dominates. Further, on the elbow and on the TCP lumped elbow mass $m_2$ and lumped tip body mass $m_3$ are assumed respectively. For this reason, the IMUs are mounted on the endpoint of each link, and we obtain

$$\mathbf{Q}_A \approx \sum_{i=2}^{3} \left(\frac{\partial_i \mathbf{v}}{\partial \dot{\mathbf{q}}_A}\right)_i^T \left[m({}_i\dot{\mathbf{v}} + {}_i\tilde{\boldsymbol{\omega}}_{Ii}\ {}_i\mathbf{v} - {}_i\mathbf{g})\right]_i$$
$$\approx m_2 \left(\frac{\partial_2 \mathbf{v}_2}{\partial \dot{\mathbf{q}}_A}\right)^T {}_2\mathbf{a}_{IMU1} + m_3 \left(\frac{\partial_3 \mathbf{v}_3}{\partial \dot{\mathbf{q}}_A}\right)^T {}_3\mathbf{a}_{IMU2}, \tag{6.19}$$

and

$$\begin{aligned} Q_{A1} &= -m_2 \cos(q_{A2}) l_2\, a_{z2} - m_3\, a_{z3}(\cos(q_{A2}) l_2 + \cos(q_{A2} + q_{A3}) l_3) \\ Q_{A2} &= m_2\, l_2\, a_{y2} + m_3(\sin(q_{A3}) l_2\, a_{x3} + (\cos(q_{A3}) l_2 + l_3) a_{y3}) \\ Q_{A3} &= m_3\, l_3\, a_{y3}. \end{aligned} \tag{6.20}$$

Based on the above findings and on (6.16), this problem can be solved by starting with the last line. In summary, the arm positions $\mathbf{q}_A$ can be computed as a function of the motor positions and the acceleration measurements at the elbow and the TCP. In order to obtain the arm velocities, only the analog rate sensors are considered. The arm velocities $\dot{\mathbf{q}}_A$ are included linearly in ${}_i\boldsymbol{\omega}_{IMU,i-1}$. Therefore we yield

$$\boldsymbol{\omega}_{IMU} = \begin{pmatrix} {}_2\boldsymbol{\omega}_{IMU1} \\ {}_3\boldsymbol{\omega}_{IMU2} \end{pmatrix} = \left(\frac{\partial \boldsymbol{\omega}_{IMU}}{\partial \dot{\mathbf{q}}_A}\right) \dot{\mathbf{q}}_A = \mathbf{J}(\mathbf{q}_A)\, \dot{\mathbf{q}}_A\,. \tag{6.21}$$

With the pseudoinverse of $\mathbf{J}$, we obtain the arm velocities in a very straightforward manner.

## 6.5 Tracking Control Framework

Several solutions to the tracking problem can be found in the literature. We use a 2DoF control scheme based on the LEM: a flatness based feedforward control $\mathbf{u}_{FF}$ using $\mathbf{q}_A$ as the flat output $\mathbf{y}$ and a passivity-based tracking error feedback control $\mathbf{u}_{FB}$. For simplicity, we employ a quasi-static tracking error system obtained by linearization along trajectory $\mathbf{q}_d$ (abbreviated LTV: linear time variant system), with $\dot{\mathbf{q}}_d = \ddot{\mathbf{q}}_d = 0$ and $\dot{\mathbf{p}}_d = 0$. Thus, the tracking error dynamics in Hamilton representation can be written as

$$\dot{\mathbf{x}}_e = \begin{pmatrix} \dot{\mathbf{e}} \\ \dot{\mathbf{p}}_e \end{pmatrix} = \begin{bmatrix} 0 & \mathbf{M}_e^{-1} \\ -\mathbf{K}_P & -\mathbf{R}_m \mathbf{M}_e^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{e} \\ \mathbf{p}_e \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{B}_e \end{pmatrix} \mathbf{u}_e, \tag{6.22}$$

with $\mathbf{e} = \mathbf{q} - \mathbf{q}_d$, $\mathbf{p}_e = \mathbf{p} - \mathbf{p}_d$ and

$$\mathbf{M}_e = \begin{bmatrix} \mathbf{M}_M & 0 \\ 0 & \mathbf{M}_A \end{bmatrix}, \quad \mathbf{R}_m = \begin{bmatrix} \mathbf{D}_M & 0 \\ 0 & \mathbf{D}_A \end{bmatrix}, \quad \mathbf{K}_P = \begin{bmatrix} \mathbf{K}_A & -\mathbf{K}_A \\ -\mathbf{K}_A & \mathbf{K}_A + \mathbf{K}_G \end{bmatrix}, \tag{6.23}$$

where $\mathbf{K}_P$ contains the spring stiffness matrix $\mathbf{K}_A$ and the gravity force matrix $\mathbf{K}_G$.The following abbreviations are used: $\mathbf{D}_M = \text{diag}([d_{M1}, d_{M2}, d_{M3}])$, $\mathbf{D}_A = \text{diag}([d_{A1}, d_{A2}, d_{A3}])$, $\mathbf{K}_A = \text{diag}([k_1, k_2, k_3])$, $\mathbf{G}_e^T = (0\,\mathbf{B}_e^T)$.

## 6.6 IDA-PB Control

In this section, we focus on the control design of port-controlled Hamiltonian systems of the tracking error dynamics

$$\dot{\mathbf{x}}_e = (\mathbf{J} - \mathbf{R}) \left( \frac{\partial H_e}{\partial \mathbf{x}_e} \right)^T + \mathbf{G}_e \mathbf{u}_e, \qquad \mathbf{y}_e = \mathbf{G}_e^T \left( \frac{\partial H_e}{\partial \mathbf{x}_e} \right)^T, \tag{6.24}$$

see also (6.8). The fundamental idea is based on the formulation of a target for the closed loop systems, such as

$$\dot{\mathbf{x}}_e = (\mathbf{J}_d - \mathbf{R}_d) \left( \frac{\partial H_d}{\partial \mathbf{x}_e} \right)^T. \tag{6.25}$$

Obviously, this results in

$$\begin{aligned} \dot{\mathbf{x}}_e &= (\mathbf{J} - \mathbf{R}) \left( \frac{\partial H_e}{\partial \mathbf{x}_e} \right)^T + \mathbf{G}_e \mathbf{u}_e \\ &\stackrel{!}{=} (\mathbf{J}_d - \mathbf{R}_d) \left( \frac{\partial H_d}{\partial \mathbf{x}_e} \right)^T, \end{aligned} \tag{6.26}$$

setting the original plant of the tracking error system in (6.24) equal to the desired system in (6.25). Computing the control law directly with

$$\mathbf{G}_e \mathbf{u}_e = (\mathbf{J}_d - \mathbf{R}_d) \left( \frac{\partial H_e}{\partial \mathbf{x}_e} \right)^T - (\mathbf{J} - \mathbf{R}) \left( \frac{\partial H_d}{\partial \mathbf{x}_e} \right)^T \tag{6.27}$$

and the pseudoinverse of $\mathbf{G}_e$, by some rearranging is apparently simple. Clearly, in the case of a non-full rank matrix $\mathbf{G}_e$ (i.e. if the system is underactuated), there are general limitations, which are shown next. In accordance with the literature [14, 19], damping injection

$$\mathbf{G}_e \mathbf{u}_e = \mathbf{G}_e \mathbf{u}_I + (\mathbf{J}_d - \mathbf{R}_d) \left( \frac{\partial H_d}{\partial \mathbf{x}_e} \right)^T - (\mathbf{J} - \mathbf{R}) \left( \frac{\partial H_e}{\partial \mathbf{x}_e} \right)^T, \tag{6.28}$$

with an additional input $\mathbf{u}_I$ is introduced. Roughly speaking, a proportional feedback of the collocated output $\mathbf{y}$ with

$$\mathbf{u}_I = -\mathbf{D}_I\,\mathbf{y} = -\mathbf{D}_I\mathbf{G}_e^T\left(\frac{\partial H_d}{\partial \mathbf{x}_e}\right)^T \tag{6.29}$$

is added. Thus, we obtain

$$\mathbf{G}_e\mathbf{u}_e = \left(\mathbf{J}_d - \overline{\mathbf{R}}_d\right)\left(\frac{\partial H_d}{\partial \mathbf{x}_e}\right)^T - \left(\mathbf{J} - \mathbf{R}\right)\left(\frac{\partial H_e}{\partial \mathbf{x}_e}\right)^T \tag{6.30}$$

with the new damping matrix $\overline{\mathbf{R}}_d = \mathbf{R}_d + \mathbf{G}_e\,\mathbf{D}_I\,\mathbf{G}_e^T$. Up to this point, the approach involves numerous design parameters, such as $H_d > 0$, $\mathbf{J}_d = -\mathbf{J}_d^T$, $\mathbf{R}_d = \mathbf{R}_d^T$, $\mathbf{D}_I$ and is therefore unsuitable from the design perspective. Typically, simplification is achieved by means of the IDA-PBC, which assumes a physical representation of the desired closed loop system, with

$$H_d = \frac{1}{2}\mathbf{e}^T\mathbf{K}_d\,\mathbf{e} + \frac{1}{2}\mathbf{p}_e^T\mathbf{M}_d^{-1}\mathbf{p}_e > 0 \tag{6.31}$$

and

$$\mathbf{J}_d = -\mathbf{J}_d^T = \begin{bmatrix} 0 & \mathbf{M}_e^{-1}\mathbf{M}_d \\ -\mathbf{M}_d\mathbf{M}_e^{-1} & \mathbf{S}_{22d} \end{bmatrix},\ \mathbf{R}_d = \mathbf{R}_d^T = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{R}_{22d} + \mathbf{B}_e\mathbf{D}_I\mathbf{B}_e^T \end{bmatrix}. \tag{6.32}$$

Adding these equations and referring to (6.26) and (6.30), we can calculate

$$\begin{aligned} \dot{\mathbf{x}}_e &= \begin{bmatrix} 0 & \mathbf{E} \\ -\mathbf{E} & -\mathbf{R}_m \end{bmatrix}\begin{pmatrix} \left(\frac{\partial H_e}{\partial \mathbf{e}}\right)^T \\ \left(\frac{\partial H_e}{\partial \mathbf{p}_e}\right)^T \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{B}_e \end{pmatrix}\mathbf{u}_e \\ &\stackrel{!}{=} \begin{bmatrix} 0 & \mathbf{M}_e^{-1}\mathbf{M}_d \\ -\mathbf{M}_e^{-1}\mathbf{M}_d & \mathbf{S}_{22d} - (\mathbf{R}_{22d} + \mathbf{B}_e\mathbf{D}_I\mathbf{B}_e^T) \end{bmatrix}\begin{pmatrix} \left(\frac{\partial H_d}{\partial \mathbf{e}}\right)^T \\ \left(\frac{\partial H_d}{\partial \mathbf{p}_e}\right)^T \end{pmatrix}. \end{aligned} \tag{6.33}$$

Satisfying the first constraint – the so-called matching condition of the generalized momenta – becomes

$$\dot{\mathbf{e}} = \left(\frac{\partial H_e}{\partial \mathbf{p}_e}\right)^T = \mathbf{M}_e^{-1}\mathbf{p}_e \stackrel{!}{=} \mathbf{M}_e^{-1}\mathbf{M}_d\left(\frac{\partial H_d}{\partial \mathbf{p}_e}\right)^T = \mathbf{M}_e^{-1}\mathbf{M}_d(\mathbf{M}_d^{-1}\mathbf{p}_e) = \dot{\mathbf{e}}. \tag{6.34}$$

As already mentioned, we obtain with the left-hand annihilator $\mathbf{B}_e^{\perp}$ implying $\mathbf{B}_e^{\perp}\mathbf{B}_e = 0$ the non-actuated projected space

$$\mathbf{B}_e^{\perp}\left(\left(\frac{\partial H_e}{\partial \mathbf{e}}\right)^T + \mathbf{R}_m\left(\frac{\partial H_e}{\partial \mathbf{p}_e}\right)^T - \mathbf{M}_d\mathbf{M}_e^{-1}\left(\frac{\partial H_d}{\partial \mathbf{e}}\right)^T + \underbrace{(\mathbf{J}_{22d} - \mathbf{R}_{22d})}_{\mathbf{JR}}\left(\frac{\partial H_d}{\partial \mathbf{p}_e}\right)^T\right) = 0, \tag{6.35}$$

for an underactuated system. Commonly, these are partial differential equations and can be split for the LTV system into $\mathbf{e}$-dependent and $\mathbf{p}_e$-dependent constraints

$$\mathbf{B}_e^{\perp}(\mathbf{K}_P - \mathbf{M}_d\mathbf{M}_e^{-1}\mathbf{K}_d) = 0 \tag{6.36}$$

$$\mathbf{B}_e^{\perp}(\mathbf{R}_m + \mathbf{JR}\mathbf{M}_d^{-1}\mathbf{M}_e) = 0, \tag{6.37}$$

and results in a coefficient comparison. Nevertheless, proof of the stability remain outstanding and implies for $H_d > 0$ (viz. $\mathbf{M}_d$, $\mathbf{K}_d > 0$) in

$$\begin{aligned}\dot{H}_d &= -\left(\frac{\partial H_d}{\partial \mathbf{p}_e}\right)\left(\mathbf{R}_{22d} + \mathbf{B}_e\mathbf{D}_I\mathbf{B}_e^T\right)\left(\frac{\partial H_d}{\partial \mathbf{p}_e}\right)^T \\ &= -\dot{\mathbf{e}}^T\mathbf{M}_e\mathbf{M}_d^{-1}\left(\mathbf{R}_{22d} + \mathbf{B}_e\mathbf{D}_I\mathbf{B}_e^T\right)\mathbf{M}_d^{-1}\mathbf{M}_e\dot{\mathbf{e}} \le 0,\end{aligned} \tag{6.38}$$

if $\mathbf{R}_{22d} + \mathbf{B}_e\mathbf{D}_I\mathbf{B}_e^T \ge 0$. Choosing the design parameters is complex and not intuitive. For the elastic robot we use

$$\begin{aligned}\mathbf{M}_d &= \begin{bmatrix}\mathbf{K}_1\mathbf{M}_M & 0\\ 0 & \mathbf{K}_2^{-1}\mathbf{M}_A(\mathbf{q}_{A,d})\end{bmatrix}, \\ \mathbf{K}_d &= \begin{bmatrix}\mathbf{K}_3 & -\mathbf{K}_A\mathbf{K}_2\\ -\mathbf{K}_2\mathbf{K}_A & \mathbf{K}_2(\mathbf{K}_A + \mathbf{K}_G)\end{bmatrix},\end{aligned} \tag{6.39}$$

and

$$\mathbf{J}_{22d} = \begin{bmatrix}0 & -\frac{1}{2}\mathbf{D}_2\\ \frac{1}{2}\mathbf{D}_2 & 0\end{bmatrix}, \quad \mathbf{R}_{22d} = \begin{bmatrix}0 & \frac{1}{2}\mathbf{D}_2\\ \frac{1}{2}\mathbf{D}_2 & \mathbf{K}_2^{-1}\mathbf{D}_A\end{bmatrix}, \tag{6.40}$$

with $\mathbf{D}_I = \mathbf{D}_1$, where all constraints are fulfilled, with $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, \mathbf{D}_1, \mathbf{D}_2 > 0$. The structure of

$$\mathbf{M}_A = \begin{bmatrix}m_{a11} & 0 & 0\\ 0 & m_{a22} & m_{a23}\\ 0 & m_{a23} & m_{a33}\end{bmatrix} \tag{6.41}$$

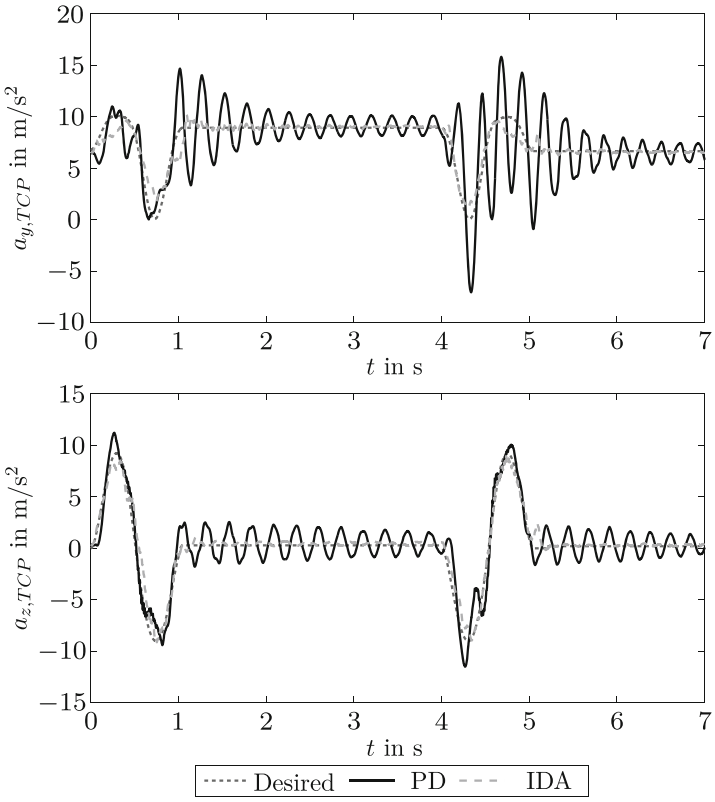



**Fig. 6.5** TCP accelerations tracking a straight line

makes satisfying the constraints in (6.23) easy by choosing two identical entries in $\mathbf{K}_2 = \text{diag}([k_{21}, k_{22}, k_{22}])$. Permutation then yields

$$\mathbf{M}_d^{-1}\mathbf{M}_e = \begin{bmatrix} \mathbf{K}_1^{-1} & 0 \\ 0 & \mathbf{K}_2 \end{bmatrix}, \quad \mathbf{M}_d\mathbf{M}_e^{-1} = \begin{bmatrix} \mathbf{K}_1 & 0 \\ 0 & \mathbf{K}_2^{-1} \end{bmatrix}, \tag{6.42}$$

which always results in a diagonal form and provides decoupled constraints. Finally, we obtain the IDA-PBC control law for the tracking error system

$$\begin{aligned} \mathbf{u}_e = \mathbf{B}_e^{+}(\mathbf{K}_A(\mathbf{e}_M - \mathbf{e}_A) - \mathbf{K}_1\,\mathbf{K}_3\,\mathbf{e}_M + \mathbf{K}_1\,\mathbf{K}_2\,\mathbf{K}_A\,\mathbf{e}_A \\ + (\mathbf{D}_M - \mathbf{B}_e^T\mathbf{D}_1\,\mathbf{B}_e\,\mathbf{K}_1^{-1})\,\dot{\mathbf{e}}_M - \mathbf{D}_2\,\mathbf{K}_2\,\dot{\mathbf{e}}_A). \end{aligned} \tag{6.43}$$

with (6.30) and the pseudo inverse $\mathbf{B}_e^{+}$. Analysis of the above equation shows that the cascaded motor joint control with very short cycle time is preserved with

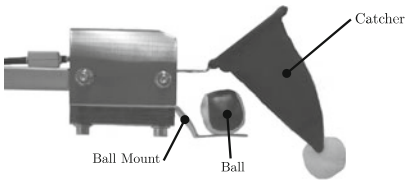



**Fig. 6.6** Ball catching setup: The ball leaves the mount as the robot moves sideways, and accelerates towards the ground

some rearranging. However, since the control concept requires knowledge of arm positions and velocities, which result from relatively complex calculations, the control scheme cannot be implemented entirely on the servo drives.

## 6.7 Experiments

As mentioned in the introduction, we verified the control strategy experimentally in a fast straight line in space and a ball catching scenario.

### *6.7.1 Straight Line in Space*

First, we tested the suggested concept for a straight line in space with the following TCP trajectory performance specifications: length approx. 2.1 m, duration 1.05 s, maximum velocity 5 m/s, and maximum acceleration 18 m/s$^2$. Figure 6.5 shows the desired TCP accelerations and those achieved with a common motor joint control (PD) and with the suggested flatness-based feedforward control in combination with the IDA-PBC feedback control design (IDA). Clearly, the PD motor joint control cannot be used for fast applications because of the high oscillations at the TCP. The proposed control concept, in contrast, suppresses vibration considerably.

### *6.7.2 Ball Catching Scenario*

In this section, we demonstrate the versatility of a lightweight structure – such as the considered elastic robot. Since we have only three joints, orientation cannot be controlled independently. Thus, we selected the setup shown in Fig. 6.6. The task of the robot was to overtake and then catch the object with the mounted hat. This process took 0.6 s from the start command to the catch and required approximately

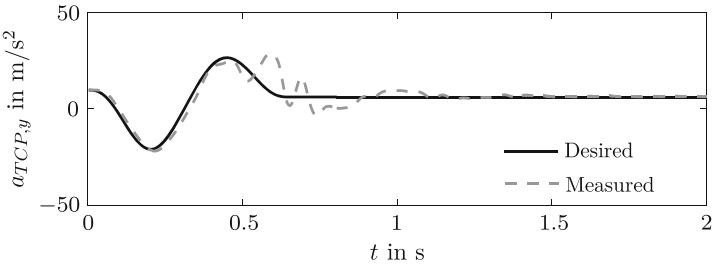



**Fig. 6.7** TCP accelerations in gravitational direction during ball catching

1.4 m in gravitational direction from start to end position. The resulting TCP acceleration in the y-direction (along which gravity is acting) is shown in Fig. 6.7. At approximately $t = 0.6$ s – when the ball is being caught – the measured acceleration deviates from the ideal. Figure 6.8 shows an image sequence of this scenario.

## 6.8 Interpretation and Conclusion

In summary, a wide range of techniques are required to handle elastic vibrations for tracking fast trajectories with satisfying accuracy. We started with a detailed model of the complex elastic multibody systems introducing Ritz Ansatz functions for the distributed parameters. Due to the complexity of the system an efficient method for modeling repeating assemblies, namely the Projection Equation in subsystem representation is used. In contrast, a relatively simple model – the LEM – is adequate for control design. Our control concept requires knowledge of arm positions and velocities. To this end, we propose a technique in which acceleration and angular rate sensors, mounted on the elbow and on the TCP can be used, to derive these system states. The Projection Equation forms the basis of this approach.

In a 2DoF control scheme, we use a flatness-based feedforward control and a passivity-based feedback control. For the latter, the IDA-PBC feedback controller was designed and resulted in a feasible feedback loop. Experimental results showed a great improvement in terms of vibration damping.

In conclusion this approach to vibration suppression in highly nonlinear systems is focused on techniques that meet the industrial demand for a lower-level implementation on common industrial hardware. We have shown that model-based control using acceleration and angular rate sensors meets this demand.

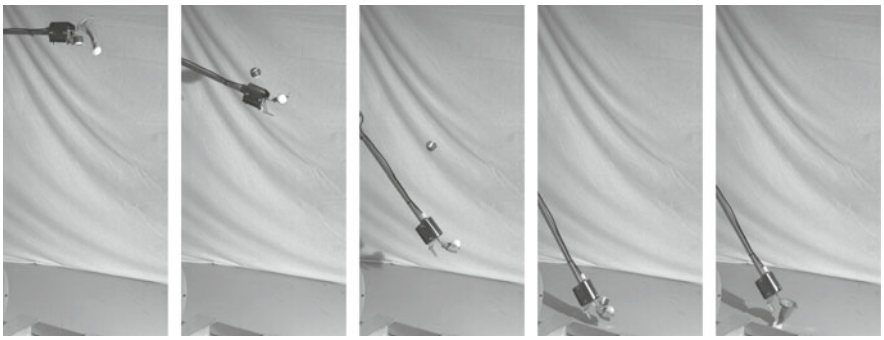



**Fig. 6.8** Image sequence of the ball catching scenario

**Acknowledgement** Support of the present work in the framework of the peer-reviewed Austrian Center of Competence in Mechatronics (ACCM) is gratefully acknowledged.

## References

1. Dwivedy SK, Eberhard P (20006) Dynamic analysis of flexible manipulators, a literature review. Mech Mach Theory 41(7):749–777
2. Bremer H (2008) Elastic multibody dynamics: a direct Ritz approach. Springer, Dordrecht
3. Calkin M (1996) Lagrangian and Hamiltonian mechanics. World Scientific, Singapore
4. Ortega R (1998) Passivity-based control of Euler-Lagrange systems: mechanical, electrical, and electromechanical applications, Communications and control engineering. Springer, London
5. Nissing D (2000) A vibration damped flexible robot: identification and parameter optimization. Am Control Conf 3:1715–1719, Chicago
6. Bernzen W (1999) Active vibration control of flexible robots using virtual spring-damper systems. J Intell Robot Syst 24:69–88
7. de Wit CC, Bastin G, Siciliano B (eds) (1996) Theory of robot control. Springer, New York
8. Wang F, Gao Y (2003) Advanced studies of flexible robotic manipulators: modeling, design, control and applications, Series in intelligent control and intelligent automation. World Scientific, New Jersey
9. Dumetz E, Dieulot JY, Barre PJ, Colas F, Delplace T (20006) Control of an industrial robot using acceleration feedback. J Intell Robot Syst 46:111–128
10. Staudecker M, Schlacher K, Hansl R (2008) Passivity based control and time optimal trajectory planning of a single mast stacker crane. In: Proceedings of the 17th world congress the international federation of automatic control, Seoul, Korea, pp 875–880
11. Höbarth W (2010) Modellierung, Steuerung und Regelung eines strukturelastischen Leichtbauroboters. PhD thesis, Johannes Kepler Universität Linz, Austria
12. Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of non-linear systems: introductory theory and examples. Int J Control 61(6):1327–1361
13. De Luca A (1996) Decoupling and feedback linearization of robots with mixed rigid/elastic joints. Proc IEEE Int Conf Robot Autom 1:816–821
14. van der Schaft AJ (2000) $L_2$-gain and passivity techniques in nonlinear control (communications and control engineering). Springer, London

15. Ortega R, van der Schaft AJ, Mareels I, Maschke B (2001) Putting energy back in control. Control Syst IEEE 21(2):18–33
16. Kugi A (2001) Non-linear control based on physical models. Springer, London
17. Gattringer H (2011) Starr-elastische Robotersysteme, Theorie und Anwendungen. Springer, Berlin
18. Kilian J, Gattringer H, Bremer H (2011) Dynamical modeling of flexible linear robots. In: Proceedings of the ASME 2011 international design engineering technical conferences computers and information in engineering conference IDETC/CIE, DETC2011/MSNDC-47442, Washington DC, USA
19. Ortega R, Schaft AJ, Maschke B, Escobar G (2002) Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. Automatica 38(4):585–596

# Chapter 7
# Norm-Optimal Iterative Learning Control for a Pneumatic Parallel Robot

**Dominik Schindele and Harald Aschemann**

**Abstract** Iterative learning control is a popular method for accurate trajectory tracking of systems that repeat the same motion many times. This paper presents a norm-optimal iterative learning control scheme for a fast two-degree-of-freedom parallel robot driven by two pairs of pneumatic muscle actuators. The robot consists of a light-weight closed-chain structure with four moving links connected by revolute joints. The two base joints are active and driven by pairs of pneumatic muscles by means of toothed belt and pulley. The proposed control has a cascade structure. The internal pressure of each pneumatic muscle is controlled by a fast underlying control loop. Hence, the control design for the outer control loop can be simplified by considering these controlled muscle pressures as ideal control inputs. The angles of the active joints as well as the corresponding angular velocities represent the controlled variables of the outer loop. The implemented ILC algorithm takes advantage of actual state information as well as of data from previous trials. Experimental results from an implementation on a test rig show an excellent control performance.

## 7.1 Introduction

Pneumatic muscles are innovative tensile actuators consisting of a fibre-reinforced vulcanised rubber tubing with appropriate connectors at both ends. The working principle is based on a rhombical fibre structure that leads to a muscle contraction in longitudinal direction when the pneumatic muscle is filled with compressed air. This contraction can be used for actuation purposes. Pneumatic muscles are low cost actuators and offer several further advantages in comparison to classical pneumatic cylinders: significantly less weight, no stick–slip effects, insensitivity

D. Schindele (✉) • H. Aschemann
University of Rostock, Rostock 18059, Germany
e-mail: dominik.schindele@uni-rostock.de; harald.aschemann@uni-rostock.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_7, 

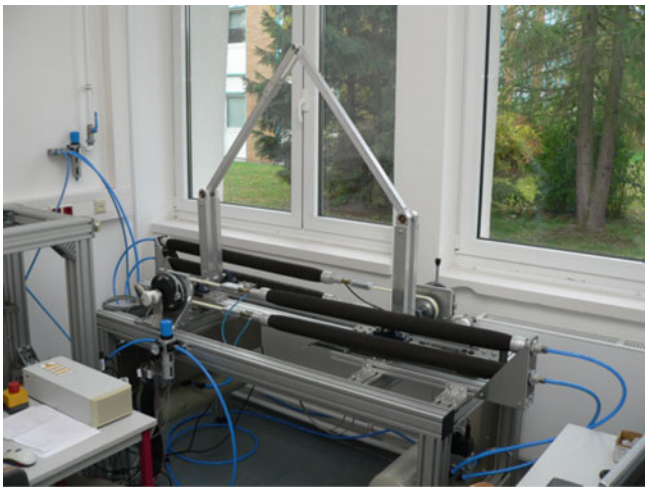


**Fig. 7.1** Test rig of the parallel robot

to dirty working environment, and a higher force-to-weight ratio. A major advantage of pneumatic drives as compared to electrical drives is their capability of providing large maximum forces for a longer period of time. In this case electrical drives are in risk of overheating and may result in increasing errors due to thermal expansion. Different kinematics actuated by pneumatic muscles have been investigated by the authors so far, such as a high-speed linear axis [5] or a delta parallel robot [13]. Applications with pneumatic muscles of other researchers are presented e.g. in [17], where a proxy-based sliding mode control for a 2-dof serial manipulator has been proposed, or in [18], where an adaptive control structure for a platform connected by a spherical joint to the base with three pneumatic muscles has been introduced.

In this contribution a norm-optimal iterative learning control (NOILC) approach is applied to a two-degree-of-freedom parallel robot actuated by four pneumatic muscles. Parallel kinematics are well known for providing high stiffness, and especially for the capability of performing fast and highly accurate motions of the end-effector. The parallel robot shown in Fig. 7.1 is characterised by a closed-chain kinematic structure formed by four moving links and the robot base. All joints are revolute joints, two of which – the cranks – are actuated by a pair of pneumatic muscles, respectively. The coordinated contraction of a pair of pneumatic muscles is transformed into a rotation of the according crank by means of a toothed belt and a pulley. The mass flow rate of compressed air into and out of a pneumatic muscle is provided by a separate proportional valve for each pneumatic muscle. The pneumatic muscles are characterised by dominant nonlinearities, namely the force characteristic and the volume characteristic.

The paper is structured as follows: first, the modelling of the mechatronic system is addressed. For the nonlinear characteristics of the pneumatic muscle, i.e. the muscle

volume and the muscle force, polynomial descriptions are used in terms of contraction length and internal muscle pressure. Second, a cascade control scheme for the parallel robot is proposed. The inner control loops involve a fast pressure control for each muscle, respectively. The outer control loop contains an iterative learning control of the two crank angles as well as the corresponding velocities as controlled variables and provides the reference pressures for the inner pressure control loops. The iterative learning control (ILC) algorithm proposed in this paper is based on a norm-optimal principle exploiting optimal feedback and feedforward actions, see [1]. Finally, the proposed control strategy has been implemented and investigated at the test rig of the Chair of Mechatronics, University of Rostock. At which, desired trajectories for the end-effector position can be tracked with high accuracy.

## 7.2 Modelling of the Parallel Robot

The modelling of the pneumatically driven parallel robot involves the mechanical subsystem and the pneumatic subsystem, which are coupled by the torques resulting from the tension forces of a pair of pneumatic muscles, respectively. The control-oriented multi-body model of the mechanical part consists of three rigid bodies (Fig. 7.2): the two cranks as actuated links with identical properties (mass $m_A$, reduced mass moment of inertia w.r.t. the actuated axis $J_{A,red}$, length of the link $l_A$, pulley radius $r$) and the end-effector $E$ (mass $m_E$), which can be modelled as lumped mass. The inertia properties of the remaining two links with length $l_P$, which are designed as light-weight construction, shall be neglected in comparison to the other links. The inertial $xz$-coordinate system is chosen in the middle of the straight line that connects both base joints with distance $2a$, as depicted in (Fig. 7.2). The motion of the parallel robot is completely described by introducing two generalised coordinates $q_1(t)$ and $q_2(t)$ that denote the two crank angles, which are combined in the vector $\boldsymbol{q} = [q_1, q_2]^T$. Analogously, the vector of the end-effector coordinates is defined as $\boldsymbol{r}_E = [x_E, z_E]^T$.

The end-effector position $\boldsymbol{r}_E$ follows for given crank angles $\boldsymbol{q}$ from the direct kinematics $\boldsymbol{r}_E = \boldsymbol{r}_E(\boldsymbol{q}, k_3)$. Each vector $\boldsymbol{q}$ allows for two different positions of the end-effector. For this reason, the configuration parameter $k_3$ as shown in Fig. 7.2 is introduced, which takes this ambiguity into account. The velocity, acceleration and the jerk of the end-effector can be obtained by subsequent differentiations

$$\dot{\boldsymbol{r}}_E = \boldsymbol{J}(\boldsymbol{q}, k_3)\dot{\boldsymbol{q}}, \tag{7.1}$$

$$\ddot{\boldsymbol{r}}_E = \boldsymbol{J}(\boldsymbol{q}, k_3)\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\dot{\boldsymbol{q}}, \tag{7.2}$$

$$\dddot{\boldsymbol{r}}_{\mathbf{E}} = \boldsymbol{J}(\boldsymbol{q}, k_3)\ \dddot{\boldsymbol{q}} + 2\dot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\ddot{\boldsymbol{q}} + \ddot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\dot{\boldsymbol{q}}, \tag{7.3}$$

with $\boldsymbol{J}(\boldsymbol{q}, k_3) = \frac{\partial \boldsymbol{r}_E}{\partial \boldsymbol{q}}$ denoting the Jacobian.

For a given end-effector position $\boldsymbol{r}_E$ the corresponding crank angles result from the inverse kinematics $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{r}_E, k_1, k_2)$. Analogue to the direct kinematics the

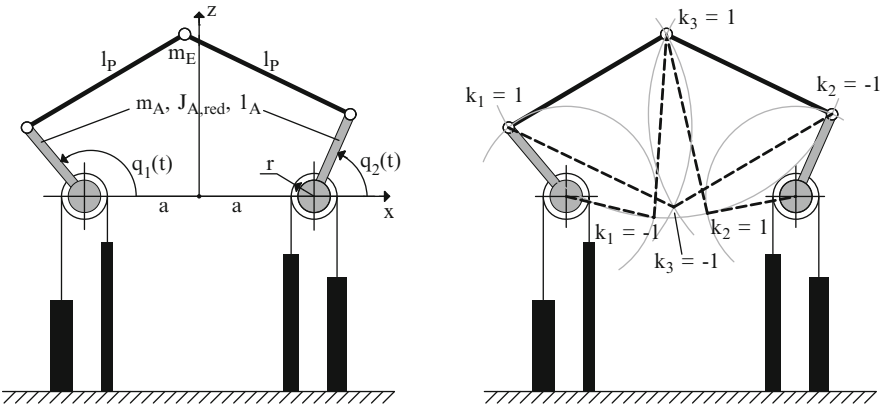



**Fig. 7.2** Two-degree-of-freedom parallel robot driven by pneumatic muscles (*left*). Ambiguity of the direct and inverse kinematics (*right*)

configuration parameters $k_1$ and $k_2$ account for the given ambiguity (Fig. 7.2). Considering the direct kinematics, the angular velocity, acceleration and jerk of the crank angles can be calculated

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}, k_3)\, \dot{\boldsymbol{r}}_E, \tag{7.4}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}, k_3)\left[\ddot{\boldsymbol{r}}_E - \dot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\dot{\boldsymbol{q}}\right], \tag{7.5}$$

$$\dddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}, k_3)\left[\dddot{\boldsymbol{r}}_E - 2\dot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{J}}(\boldsymbol{q}, k_3)\dot{\boldsymbol{q}}\right]. \tag{7.6}$$

To determine the according equations of motion for the actuated links, Lagrange's equations can be applied

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i^n, \tag{7.7}$$

where $T$ denotes the kinetic energy, $U$ the potential energy and $Q_i$ the generalised torques. Evaluating (7.7) leads to the following form of the equations of motion

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{k}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{Q}^{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{u}, \boldsymbol{z}), \tag{7.8}$$

with the symmetric mass matrix $\boldsymbol{M}(\boldsymbol{q})$, the vector of centrifugal and Coriolis terms $\boldsymbol{k}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the gravity torque vector $\boldsymbol{G}(\boldsymbol{q})$. The vector of the generalised torques $\boldsymbol{Q}^{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{u}, \boldsymbol{z}) = [\tau_1 - \tau_{U1}, \tau_2 - \tau_{U2}]^T$ consists of the drive torques $\tau_i = r\,[F_{Mil}(p_{Mil}, q_i) - F_{Mir}(p_{Mir}, q_i)]$ and the disturbance torques $\tau_{Ui}$, whereas the index $i = \{1, 2\}$ denotes the drive under consideration. The vector

of input variables is given by $\boldsymbol{u} = [p_{M1l}, p_{M1r}, p_{M2l}, p_{M2r}]^T$, the vector of disturbance torques by $\boldsymbol{z} = [\tau_{U1}, \tau_{U2}]^T$.

## 7.3 Modelling of the Pneumatic Actuators

The parallel robot is equipped with four pneumatic muscle actuators. A mass flow $\dot{m}_{Mij}$ – the index $j = \{l, r\}$ stands for the left or right pneumatic muscle of each drive – into the pneumatic muscle leads to an increase in internal pressure $p_{Mij}$, associated with a contraction $\Delta\ell_{Mij}$ of the muscle in longitudinal direction. This contraction effect can be exploited to generate forces. The contraction lengths of the pneumatic muscles are related to the generalised coordinates, i.e. the crank angles $q_i$. The position of the crank angle, where the corresponding right pneumatic muscle is fully contracted, is denoted by $q_{i0}$. Consequently, by considering the transmission consisting of toothed belt and pulley, the following constraints hold for the contraction lengths of the muscles

$$\Delta\ell_{Mil}(q_i) = r\,(q_i - q_{i0}), \tag{7.9}$$

$$\Delta\ell_{Mir}(q_i) = \Delta\ell_{M,max} - r\,(q_i - q_{i0}). \tag{7.10}$$

Here, $\Delta\ell_{M,max}$ is the maximum contraction given by 25% of the uncontracted length. The force characteristic $F_{Mij}(p_{Mij}, \Delta\ell_{Mij})$ of the pneumatic muscle yields the resulting static tension force for given internal pressure $p_{Mij}$ as well as given contraction length $\Delta\ell_{Mij}$. This nonlinear force characteristic has been identified by static measurements and, then, approximated by the following polynomial description

$$F_{Mij}(p_{Mij}, \Delta\ell_{Mij}) = \begin{cases} \bar{F}_{Mij}\,(p_{Mij}, \Delta\ell_{Mij}) & \text{if } \bar{F}_{Mij} > 0 \\ 0 & \text{else} \end{cases}, \tag{7.11}$$

with

$$\bar{F}_{Mij} = \sum_{m=0}^{3} \left(a_m\,\Delta\ell_{Mij}^m\right) p_{Mij} - \sum_{n=0}^{4} \left(b_n\,\Delta\ell_{Mij}^n\right). \tag{7.12}$$

As the maximum internal pressure is limited by a maximal value of $p_{Mij,max} = 7$ *bar*, the ideal gas equation $p_{Mij} = \rho_{Mij} R T_{Mij}$ can be utilised as accurate description of the thermodynamic behaviour of the compressed air. Here, the density $\rho_{Mij}$, the gas constant $R$, and the thermodynamic temperature $T_{Mij}$ are introduced. The thermodynamic process can be described as a polytropic change of state $p_{Mij}/\rho_{Mij}^n = const.$, where $n = 1.26$ denotes the identified polytropic exponent.

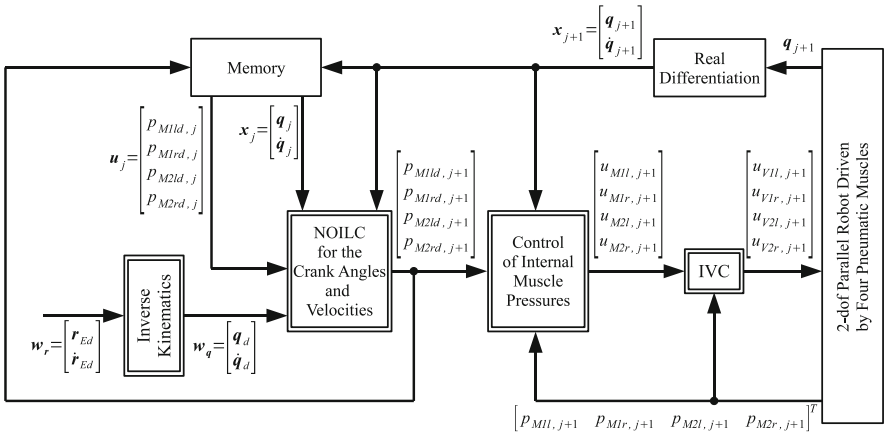



**Fig. 7.3** Cascaded control structure

The volume characteristic of the pneumatic muscle can be accurately approximated by the polynomial function

$$V_{Mij}\left(\Delta\ell_{Mij}, p_{Mij}\right) = \sum_{k=0}^{3}\left(a_k\,\Delta\ell_{Mij}^{k}\right)p_{Mij} + \sum_{l=0}^{3} b_l\,\Delta\ell_{Mij}^{l}, \tag{7.13}$$

where the coefficients $a_k$ and $b_l$ have been identified by measurements. Finally, the state equation for the internal muscle pressure follows directly from a mass flow balance in combination with the pressure-density relationship [4]

$$\dot{p}_{Mij} = \frac{n}{V_{Mij} + n\frac{\partial V_{Mij}}{\partial p_{Mij}}p_{Mij}}\left[u_{Mij} - \frac{\partial V_{Mij}}{\partial \Delta\ell_{Mij}}\frac{\mathrm{d}\Delta\ell_{Mij}}{\mathrm{d}q_i}\,p_{Mij}\,\dot{q}_i\right], \tag{7.14}$$

where $u_{Mij} = RT_{Mij}\,\dot{m}_{Mij}$ denotes the input variable. Usually, the temperature $T_{Mij}$ is not measured but approximated by the constant temperature $T_0$ of the ambiance [8].

## 7.4 Cascaded Control Structure

For control design a cascaded structure has been chosen. The mechanical subsystem (7.8) is controlled by an outer control loop, whereas the pneumatic subsystem (7.14) is controlled by an underlying control loop. In this way the complexity at control design is reduced and a fast control of the internal muscle pressures is ensured. The proposed control structure is depicted in Fig. 7.3. The nonlinear characteristic of the proportional valves has beenidentified by measurements, see [5], and is compensated by its approximated inverse valve characteristic (IVC) in each input

channel. Hence, the variable $u_{Mij}$ can be used as new control input for the underlying pressure control. For control design of the internal muscle pressures the differential flatness of the pneumatic subsystem is exploited to compensate all nonlinearities and a control Lyapunov function is designed to stabilise the dynamics of the tracking error $e_{ij} = p_{Mijd} - p_{Mij}$, cf. [15] for details. The control design for the outer control loop can be simplified by considering the internal muscle pressures $p_{Mij}$ as ideal control inputs. The controlled variables of the outer loop are represented by the crank angles $q_i$ as well as the corresponding velocities $\dot{q}_i$. For this control loop a norm-optimal iterative learning control approach is employed as proposed in [1] or [14]. For this purpose the tracking error of the $k$-th trial is stored in the memory and is utilised by the ILC controller to improve the control performance of the $(k+1)$-th trial. Thus, remaining model uncertainties can be compensated iteratively.

## 7.5 Norm-Optimal Iterative Learning Control

ILC is a method for improving the tracking behaviour of repetitive motion tasks. For this purpose the tracking error of the actual trial is employed to improve the control behaviour for the future trials [9]. Repetitive tasks can be found in many industrial applications, such as automated manufacturing systems, chemical processes or robotics. Since iterative learning control was introduced by [3], a lot of different algorithms have been derived. An overview of ILC is presented in [7] or [10]. Several control disciplines have had an influence on the area of ILC, such as the optimal control theory, see e.g. [16]. The norm-optimal iterative learning control (NOILC) employed in this paper is described in detail in [1]. Often, ILC is a pure feedforward control strategy. In contrast, the ILC algorithm presented in this contribution can be considered as a combination of previous cycle feedback, which can be interpreted as a feedforward action, and current cycle feedback. For the control design a discrete-time system representation in the following form is considered

$$\begin{aligned} \boldsymbol{x}_j(k+1) &= \boldsymbol{f}\big(\boldsymbol{x}_j(k), \boldsymbol{u}_j(k)\big) = \boldsymbol{A}\boldsymbol{x}_j(k) + \boldsymbol{B}\boldsymbol{u}_j(k), \\ \boldsymbol{y}_j(k) &= \boldsymbol{h}\big(\boldsymbol{x}_j(k)\big) = \boldsymbol{C}\boldsymbol{x}_j(k) \end{aligned} \tag{7.15}$$

with the state vector $\boldsymbol{x}_j(k)$, the input vector $\boldsymbol{u}_j(k)$ and the output vector $\boldsymbol{y}_j(k)$. The index $j$ denotes the current trial and $k \in \mathbb{N}_0$, $(0 \leq k \leq N)$ stands for the discrete-time index. Introducing the super-vectors[1] $\boldsymbol{y} = \Big[\boldsymbol{y}_j^T(1), \boldsymbol{y}_j^T(2), \cdots, \boldsymbol{y}_j^T(N)\Big]^T$ and $\boldsymbol{u} = \Big[\boldsymbol{u}_j^T(0), \boldsymbol{u}_j^T(1), \cdots, \boldsymbol{u}_j^T(N-1)\Big]^T$ as well as the matrix $\boldsymbol{G}$ as

[1] The super-vectors are marked by the omission of the time argument.

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{CB} & 0 & \cdots & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{N-1}\boldsymbol{B} & \boldsymbol{CA}^{N-2}\boldsymbol{B} & \cdots & \boldsymbol{CB} \end{bmatrix}, \tag{7.16}$$

the system (7.15) can be stated in the form

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{u}, \tag{7.17}$$

for vanishing initial conditions. Then for a desired reference trajectory $\boldsymbol{y}_d$ the tracking error is given by

$$\boldsymbol{e} = \boldsymbol{y}_d - \boldsymbol{y} = \boldsymbol{y}_d - \boldsymbol{G}\boldsymbol{u}. \tag{7.18}$$

The problem to be solved now is to determine the optimal solution $\boldsymbol{u}^*$ of the minimisation problem

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \|\boldsymbol{e}\|^2 = \arg\min_{\boldsymbol{u}} \left\|\boldsymbol{y}_d - \boldsymbol{G}\,\boldsymbol{u}\right\|^2. \tag{7.19}$$

In the ideal case, when the design model matches the real system exactly, the solution of (7.19) can be found in one step $\boldsymbol{u}^* = \boldsymbol{G}^{-1}\boldsymbol{y}_d$ if $\boldsymbol{G}$ is invertible. In the given case of an imperfect system model with remaining uncertainties and disturbances, however, an iterative solution is needed, which is performed on the plant during its operation as an experimental procedure here. Thus, an algorithm is considered, which aims at reducing the norm of the tracking error at each iteration step $j_k$. For this purpose the norm-optimal problem, which has to be solved for the iterated inputs $\boldsymbol{u}_{j+1}$, is formulated as

$$\boldsymbol{u}_{j+1} = \arg\min_{\boldsymbol{u}_{j+1}} J_{j+1}\left(\boldsymbol{u}_{j+1}\right), \tag{7.20}$$

with a quadratic cost function $J_{k+1}$ according to

$$J_{j+1} = \frac{1}{2}\left[\boldsymbol{e}_{j+1}^T\boldsymbol{Q}\boldsymbol{e}_{j+1} + \left(\boldsymbol{u}_{j+1} - \boldsymbol{u}_j\right)^T\boldsymbol{R}\left(\boldsymbol{u}_{j+1} - \boldsymbol{u}_j\right)\right]. \tag{7.21}$$

Considering the necessary optimality condition

$$\frac{\partial J_{j+1}}{\partial \boldsymbol{u}_{j+1}} = -\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{e}_{j+1} + \boldsymbol{R}\left(\boldsymbol{u}_{j+1} - \boldsymbol{u}_j\right) \stackrel{!}{=} \boldsymbol{0}, \tag{7.22}$$

the optimal control input results in

$$\boldsymbol{u}_{j+1} = \boldsymbol{u}_j + \boldsymbol{R}^{-1}\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{e}_{j+1}. \tag{7.23}$$

This solution, however, is anti-causal. To obtain a causal solution, the cost function can be reformulated as follows

$$\begin{aligned} J_{j+1} = &\frac{1}{2}\Bigg[\sum_{k=1}^{N} \boldsymbol{e}_{j+1}^{T}(k)\boldsymbol{Q}(k)\boldsymbol{e}_{j+1}(k) \\ &+ \sum_{k=0}^{N-1} \big(\boldsymbol{u}_{j+1}(k) - \boldsymbol{u}_{j}(k)\big)^{T} \boldsymbol{R}(k)\big(\boldsymbol{u}_{j+1}(k) - \boldsymbol{u}_{j}(k)\big)\Bigg] \\ = &\, h_0\big(\boldsymbol{x}_{j+1}(N)\big) + \sum_{k=0}^{N-1} f_0\big(\boldsymbol{x}_{j+1}(k), \boldsymbol{u}_{j+1}(k), \boldsymbol{u}_{j}(k)\big), \end{aligned} \tag{7.24}$$

with the weighting matrices $\boldsymbol{Q}(k)$ and $\boldsymbol{R}(k)$ are chosen symmetric and positive definite. For solving the given optimisation problem, the maximum principle of Pontryagin is employed. By utilising the Hamiltonian $H_{j+1}$ with the adjoint states $\psi_j(k)$

$$H_{j+1} = -f_0\big(\boldsymbol{x}_{j+1}(k), \boldsymbol{u}_{j+1}(k), \boldsymbol{u}_{j}(k)\big) + \boldsymbol{\psi}_{j+1}^{T}(k+1)\, \boldsymbol{f}\big(\boldsymbol{x}_{j+1}(k), \boldsymbol{u}_{j+1}(k)\big) \tag{7.25}$$

the origin system can be described by the following costate system, cf. [6]

$$\boldsymbol{x}_{j+1}(k+1) = \frac{\partial H_{j+1}}{\partial \boldsymbol{\psi}_{j+1}(k+1)} = \boldsymbol{A}\boldsymbol{x}_{j+1}(k) + \boldsymbol{B}\boldsymbol{u}_{j+1}(k), \tag{7.26a}$$

$$\boldsymbol{\psi}_{j+1}(k) = \frac{\partial H_{j+1}}{\partial \boldsymbol{x}_{j+1}(k)} = \boldsymbol{A}^{T}\boldsymbol{\psi}_{j+1}(k+1) + \boldsymbol{C}^{T}\boldsymbol{Q}(k)\boldsymbol{e}_{j+1}(k), \tag{7.26b}$$

with the terminal condition $\boldsymbol{\psi}_{j+1}(N) = -\frac{\partial h_0}{\partial x_{j+1}(N)} = \boldsymbol{C}^{T}\boldsymbol{Q}(N)\boldsymbol{e}_{j+1}(N)$. Further, the necessary optimality condition

$$\frac{\partial H_{j+1}}{\partial \boldsymbol{u}_{j+1}(k)} = \boldsymbol{0} = \boldsymbol{R}(k)\big(\boldsymbol{u}_{j}(k) - \boldsymbol{u}_{j+1}(k)\big) + \boldsymbol{B}^{T}\boldsymbol{\psi}_{j+1}(k+1) \tag{7.27}$$

leads to

$$\boldsymbol{u}_{j+1}(k) = \boldsymbol{u}_{j}(k) + \boldsymbol{R}^{-1}(k)\boldsymbol{B}^{T}\boldsymbol{\psi}_{j+1}(k+1). \tag{7.28}$$

Under the assumption of a full state knowledge, the following ansatz for the adjoint equations is made

$$\boldsymbol{\psi}_{j+1}(k) = -\boldsymbol{K}(k)\big(\boldsymbol{x}_{j+1}(k) - \boldsymbol{x}_{j}(k)\big) + \boldsymbol{\xi}_{j+1}(k). \tag{7.29}$$

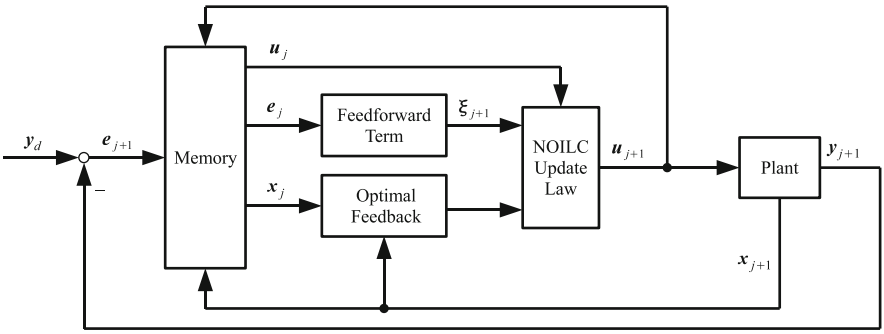



**Fig. 7.4** Block diagram of the NOILC-algorithm

By inserting (7.29) in (7.26), it can be shown, that the postulated solution (7.29) and the formal solution are equal, if $\boldsymbol{K}(k)$ and $\boldsymbol{\xi}_{j+1}(k)$ are satisfying specific equations. The matrix gain $\boldsymbol{K}(k)$ can be calculated in descending sample order as solution of the discrete matrix Riccati equation

$$\begin{aligned}\boldsymbol{K}(k) = {} & \boldsymbol{A}^T\boldsymbol{K}(k+1)\boldsymbol{A} + \boldsymbol{C}^T\boldsymbol{Q}(k+1)\boldsymbol{C} - \boldsymbol{A}^T\boldsymbol{K}(k+1)\boldsymbol{B} \\ & \times \left(\boldsymbol{B}^T\boldsymbol{K}(k+1)\boldsymbol{B} + \boldsymbol{R}(k+1)\right)^{-1}\boldsymbol{B}^T\boldsymbol{K}(k+1)\boldsymbol{A}, \end{aligned} \tag{7.30}$$

with the terminal condition $\boldsymbol{K}(N) = \boldsymbol{C}^T\boldsymbol{Q}(N)\boldsymbol{C}$. The feedforward term $\boldsymbol{\xi}_{j+1}(k)$, depending on the tracking error of the previous trial, is also computed in descending sample order

$$\begin{aligned}\boldsymbol{\xi}_{j+1}(k) = {} & \boldsymbol{A}^T\left[\boldsymbol{I} - \boldsymbol{K}(k+1)\boldsymbol{B}\left(\boldsymbol{B}^T\boldsymbol{K}(k+1)\boldsymbol{B} + \boldsymbol{R}(k+1)\right)^{-1}\boldsymbol{B}^T\right] \\ & \times \boldsymbol{\xi}_{j+1}(k+1) + \boldsymbol{C}^T\boldsymbol{Q}(k)\boldsymbol{e}_j(k), \end{aligned} \tag{7.31}$$

with the terminal condition $\boldsymbol{\xi}_{j+1}(N) = \boldsymbol{C}^T\boldsymbol{Q}(N)\boldsymbol{e}_j(N)$. Finally, the update law for the control input results from (7.28) and (7.26)

$$\begin{aligned}\boldsymbol{u}_{j+1}(k) = {} & \boldsymbol{u}_j(k) - \left(\boldsymbol{B}^T\boldsymbol{K}(k+1)\boldsymbol{B} + \boldsymbol{R}(k+1)\right)^{-1}\boldsymbol{B}^T \\ & \times \left[\boldsymbol{K}(k+1)\boldsymbol{A}\left(\boldsymbol{x}_{j+1}(k) - \boldsymbol{x}_j(k)\right) + \boldsymbol{\xi}_{j+1}(k+1)\right]. \end{aligned} \tag{7.32}$$

The block diagram of the NOILC-algorithm is depicted in Fig. 7.4. Including current cycle feedback in the ILC control law (7.32) results in an improved robustness. Regarding the implementation, the matrix gain $\boldsymbol{K}(k)$ has to be calculated only once at the initialisation of the ILC algorithm. The evaluation of (7.31), which depends on the control error of the previous trial, can be performed after the end of every trial. Only the equation for the input update (7.32), which involves both feedback and feedforward actions, have to be evaluated during each sample

time step. In [1] and [11], it is shown that the algorithm converges exponentially with the following convergence rate

$$\left\|\boldsymbol{e}_{j+1}\right\| \leq \frac{1}{1+\sigma^2}\left\|\boldsymbol{e}_j\right\|, \tag{7.33}$$

with $\sigma$ satisfying the condition $\boldsymbol{G}\boldsymbol{R}^{-1}\boldsymbol{G}^T\boldsymbol{Q} \geq \sigma^2\boldsymbol{I}$. As a result the convergence rate can be affected by the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. The presented NOILC algorithm supposes a full knowledge of the system states. This implies that all system states are either measurable or can be observed. A similar algorithm for continuous time systems is presented in [2].

## 7.6 NOILC Applied to the Parallel Robot

As basis for the NOILC approach, the state-space representation of a linear discrete-time system is needed. Hence, a linearisation and a subsequent discretisation of the state-space equation

$$\begin{bmatrix}\dot{\boldsymbol{q}}\\ \ddot{\boldsymbol{q}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}}\\ \boldsymbol{M}^{-1}(\boldsymbol{q})(\boldsymbol{Q}^n(\boldsymbol{q},\boldsymbol{u},\boldsymbol{z}) - \boldsymbol{k}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{G}(\boldsymbol{q}))\end{bmatrix}, \quad \boldsymbol{y} = \boldsymbol{C}\,\boldsymbol{x} = \boldsymbol{x}, \tag{7.34}$$

with $\boldsymbol{x} = [\boldsymbol{q}, \dot{\boldsymbol{q}}]^T$, $\boldsymbol{u} = [p_{M1l}, p_{M1r}, p_{M2l}, p_{M2r}]^T$ is mandatory. For this purpose the system is linearised along a desired trajectory using a first-order Taylor series expansion, whereas the explicit Euler method is employed for time discretisation. To further reduce the computational effort at each time step, constant terms appearing in the ILC algorithm can be combined and calculated only once at initialisation of the algorithm. Such an approach is also presented in [12] as fast NOILC (F-NOILC). As the original system model of the high-speed linear axis is nonlinear, the error made by a linearisation of the system as well as the error resulting from disregarding model uncertainties and disturbances $\boldsymbol{z}$ have to be compensated by the learning law. The proposed algorithm achieves good convergence and yields excellent tracking performance as shown by experimental results in Sect. 7.7 even though the controlled system is nonlinear.

## 7.7 Experimental Results

Both tracking performance and steady-state accuracy w.r.t. the end-effector position $\boldsymbol{r}_E$ have been investigated by experiments at the test rig of the Chair of Mechatronics, University of Rostock. It is equipped with four pneumatic muscles. The internal pressures of the muscles are measured by piezo-resistive pressure

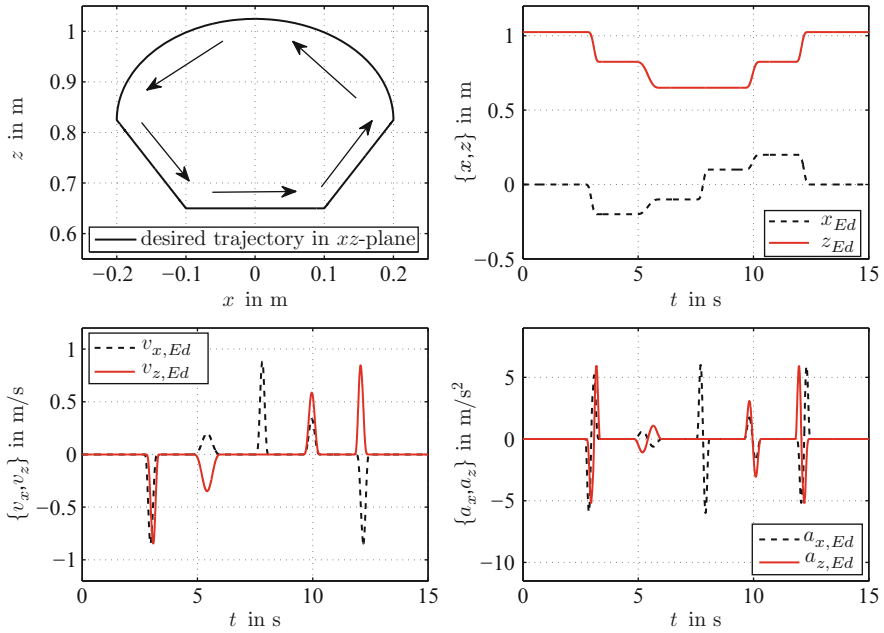



**Fig. 7.5** Desired trajectories in the workspace

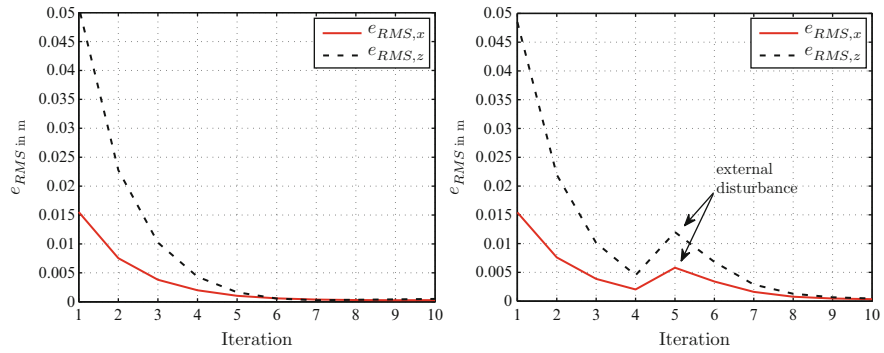



**Fig. 7.6** *Left part*: Root mean square error of the $x$- and $z$-axis for ten iterations. *Right part*: RMS error with an external disturbance at iteration 5

sensors. The crank angles $q_i$ are obtained by two rotary encoders providing high resolution. The control algorithm has been implemented on a dSpace real-time system with a sample rate of 1 ms. The desired value for the time derivative of the internal muscle pressure is obtained by a model-based calculation using only desired values, i.e.

$$\dot{p}_{Mijd} = \dot{p}_{Mijd}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \dddot{\boldsymbol{q}}, p_{Mid}, \dot{p}_{Mid}). \tag{7.35}$$

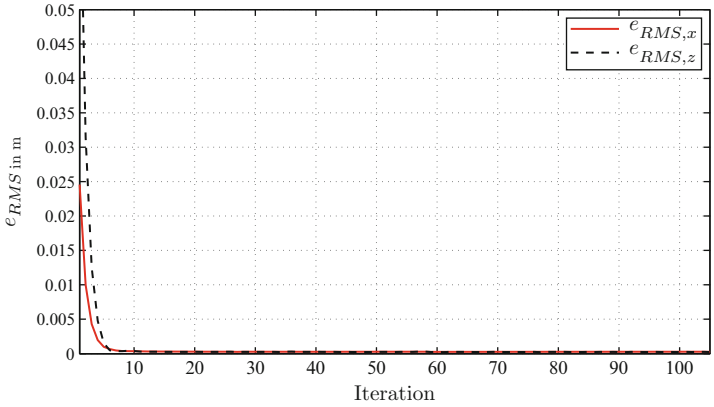



**Fig. 7.7** Long-term performance of the NOILC approach

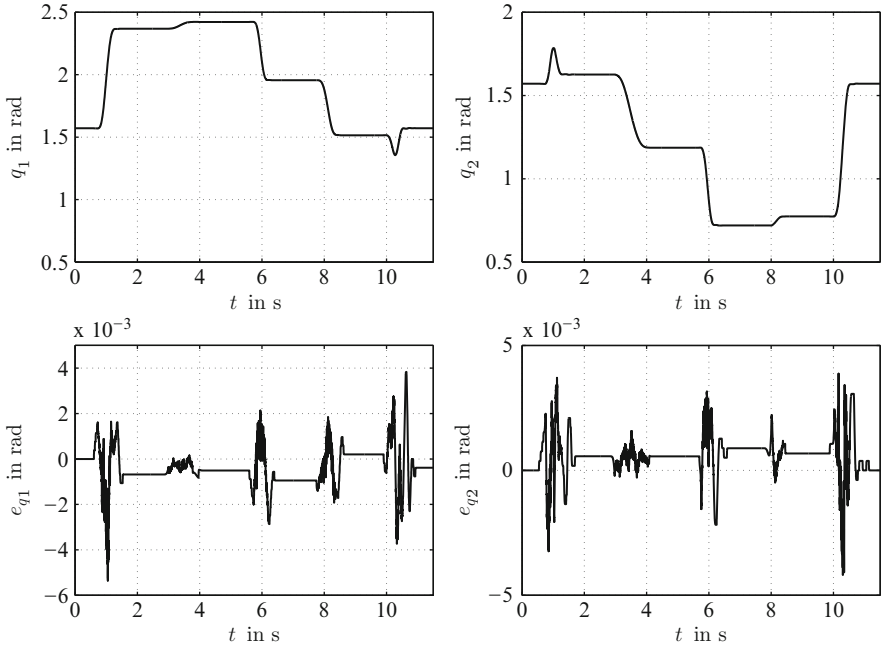



**Fig. 7.8** Tracking errors of the controlled variables for one cycle

Here $p_{Mid}$ and $\dot{p}_{Mid}$ denote the desired mean pressure and its time derivative for the left and the right drive. The desired trajectories for the end-effector position and its corresponding time derivatives are obtained from a trajectory planning module that provides synchronous time optimal trajectories. For the experiments the trajectory shown in Fig. 7.5 have been used. The maximum velocities are approx. 0.9 m/s

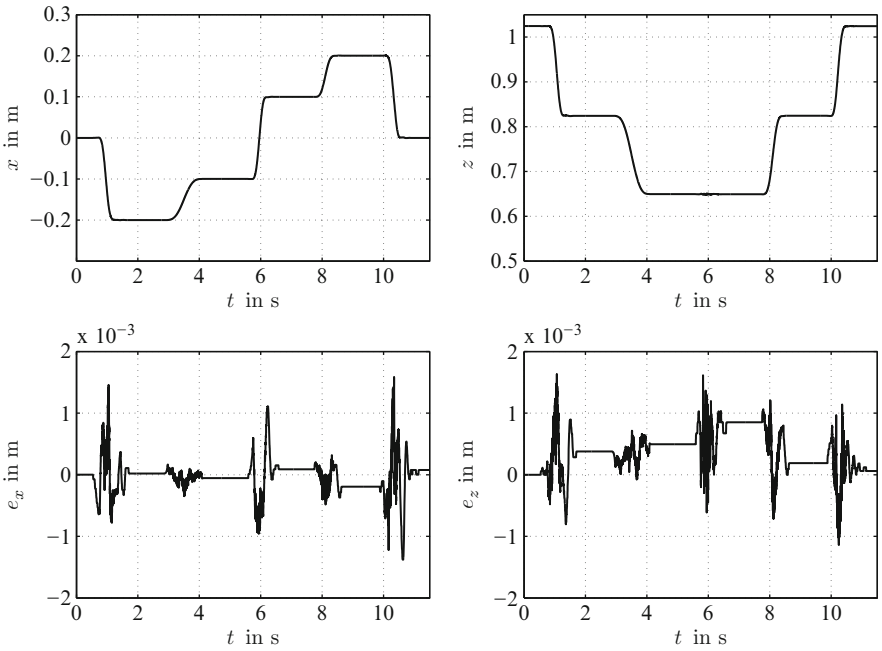



**Fig. 7.9** Tracking errors in the workspace for one cycle

and the maximum accelerations are approx. 5 m/s$^2$. In Fig. 7.6, the root-mean square (RMS) errors

$$e_{RMS,x} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(x_d(k) - x(k)\right)^2}, \quad e_{RMS,z} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(z_d(k) - z(k)\right)^2} \tag{7.36}$$

are depicted for one run with ten iterations. Already after three cycles the RMS error of both axes is below 0.5 mm. The robustness of the control algorithm is investigated by applying an external disturbance to the robot, as shown in the right part of Fig. 7.6. The system remains stable, and after only a few iterations the RMS error is as small as before. To achieve also a good long-term performance the signals $\boldsymbol{u}_k$ are filtered offline between the separate trials by a low-pass filter with zero-phase distortion. Thus, unreproducible effects are faded out and an increase of the error after several iterations is avoided. An experiment with 100 iterations, see Fig. 7.7, shows that the system remains stable also for a large number of iterations, and the RMS error remains bounded. The chronological sequence of the tracking errors of the controlled variables $e_{q1}$ and $e_{q2}$ are depicted in Fig. 7.8, whereas the tracking errors in the workspace $e_x$ and $e_z$ are shown in Fig. 7.9 for one trial in each case. Here, the errors were recorded during the tenth iteration. As can be seen, the maximum error in the workspace during the acceleration and deceleration intervals

is approx. 1.5 mm, whereas the maximum steady-state error is below 0.9 mm. Hence, the tracking behaviour is significantly improved as compared to non-learning model-based control approaches, such as sliding-mode control, cf. [15].

## 7.8 Conclusion

In this paper an iterative learning control approach for trajectory control is presented for a 2-dof parallel robot driven by pneumatic muscles. The modelling of this mechatronic system leads to a system of eight nonlinear first-order differential equations. For the nonlinear characteristics of the pneumatic muscles polynomials serve as good approximations. The control has a cascaded structure: while the internal muscle pressures represent the controlled variables for the inner control loop, the crank angles and the corresponding velocities are controlled in the decoupling outer loop. For the outer control loop a norm-optimal learning controller was designed. The implemented NOILC algorithm uses informations of the state variables of the current trial as well as the previous trial and offer exponential convergence rates. Experimental results emphasize the excellent closed-loop performance with maximum position errors of approx.1.5 mm.

## References

1. Amann N, Owens DH, Rogers E (1996) Iterative learning control for discrete-time systems with exponential rate of convergence. IEEE Proc: Control Theory Appl 143(2):217–224
2. Amann N, Owens DH, Rogers E (1996) Iterative learning control using optimal feedback and feedforward actions. Int J Control 65(2):277–293
3. Arimoto S, Kawamura S, Miyazaki F (1984) Bettering operation of robots by learning. J Robot Syst 1:123–140
4. Aschemann H, Schindele D (2008) Nonlinear model-based control of a parallel robot driven by pneumatic muscle actuators. In: Aschemann H (ed) New approaches in automation and robotics. I-Tech Education and Publishing, Vienna, pp 25–40
5. Aschemann H, Schindele D (2008) Sliding-mode control of a high-speed linear axis driven by pneumatic muscle actuators. IEEE Trans Ind Electron 55(11):3855–3864
6. Athans M, Falb P (1966) Optimal control. McGraw-Hill, New York
7. Bristow D, Tharayil M, Alleyne A (2006) A survey of iterative learning control. IEEE Contr Syst Mag 26(3):96–114
8. Göttert M (2004) Bahnregelung servopneumatischer Antriebe. Berichte aus der Steuerungs- und Regelungstechnik (in German). Shaker Verlag, Aachen
9. Longman RW (2000) Iterative learning control and repetitive control for engineering practice. Int J Control 73:930–954
10. Moore K (1999) Iterative learning control: an expository overview. Appl Comput Control, Signal Circuit 1:151–214
11. Owens D, Daley S (2008) Iterative learning control – monotonicity and optimization. Int J Appl Math Comput Sci 18(3):279–293

12. Ratcliffe J, Lewin P, Rogers E, Hätönen J, Owens D (2006) Norm-optimal iterative learning control applied to gantry robots for automation applications. IEEE Trans Robot 22 (6):1303–1307
13. Schindele D, Aschemann H (2009) Adaptive backstepping control of a delta parallel robot driven by pneumatic muscle actuators. In: Proceedings of 15th international workshop on dynamics and control IWDC, Tossa de Mar, pp 151–158
14. Schindele D, Aschemann H (2010) Norm-optimal iterative learning control for a highspeed linear axis with pneumatic muscles. In: Proceedings of NOLCOS 2010, Bologna, pp 463–468
15. Schindele D, Aschemann H (2010) Trajectory tracking of a pneumatically driven parallel robot using higher-order SMC. In: Proceedings of MMAR 2010, Miedzyzdroje, pp 387–392
16. Schöllig A, D'Andrea R (2009) Optimization-based iterative learning control for trajectory tracking. In: Proceedings of European control conference 2009, Budapest, pp 1505–1510
17. Van-Damme M, Vanderborght R, Ham RV, Verrelst B, Daerden F, Lefeber D (20007) Proxy-based sliding-mode control of a manipulator actuated by pleated pneumatic artificial muscles. In: Proceedings of IEEE international conference on robotics and automation, Rome, pp 4355–4360
18. Zhu X, Tao G, Yao B, Cao J (2008) Adaptive robust posture control of a parallel manipulator driven by pneumatic muscles. Automatica 44(9):2248–2257

# Chapter 8
# Balance and Posture Control for Biped Robots

**Maximo A. Roa and Christian Ott**

**Abstract** This work presents an overview of a new approach for balance and posture control by regulating simultaneously the center of mass position and trunk orientation of a biped robot. After an unknown external perturbation deviates the robot from a desired posture, the controller computes a wrench (force and torque) required to recover the desired position and orientation, according to a compliance control law. This wrench is distributed to predefined supporting contact points at the feet. The forces at these points are computed via a constrained optimization problem, adopted from the grasping literature, which minimizes the contact forces while including friction restrictions and torque limits at each joint.

## 8.1 Introduction

The goal of obtaining biped robots able to interact with humans in everyday tasks and environments, calls for a proper control system that allows the robot to balance (compliantly) in the presence of unknown external perturbations. Such balance, i.e. the control of the linear and angular momentum of the system, is achieved through the application of suitable contact forces to the ground, using the finite support area of the feet [12]. Traditional approaches use a dynamics based walking pattern generator that provides desired trajectories for the underlying position controllers. The execution of such trajectories requires the addition of force sensors in the feet for implementing a Zero Moment Point (ZMP) control loop. In this way, a large range of stepping and walking motions can be generated. Several position-based balance compensators have been developed, although in general they require the measurement of force at every expected point of interaction with the external environment, which increases the computational load and creates time delays in the controller [1, 7, 20, 22].

M.A. Roa (✉) • C. Ott
Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany
e-mail: Maximo.Roa@dlr.de; Christian.Ott@dlr.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_8, 

**Fig. 8.1** DLR-Biped: a biped walking machine with torque controlled joints

Joint torque sensing and control allows sensitive compliance and impedance control [14], but requires additional instrumentation in the drive units. Torque sensing has been applied explicitly in the hydraulic humanoid robot CB [3] and implicitly via serial elastic actuators in [17]. At DLR, joint torque sensors are integrated in an electrically driven biped robot based on the torque controlled drive units of the DLR-KUKA Light-Weight Robot (Fig. 8.1), which can be position or torque controlled [15].

Passivity-based impedance and compliance controllers based on joint torque sensing have been traditionally applied to manipulation tasks [1, 14]. The application of such framework to biped balancing control was first proposed in [9]. This controller provides gravity compensation, making the robot compliant and thus facilitating physical interactions and adaptation to unknown external forces. It sets a ground reaction force able to compensate perturbations on the robot position, and transforms the desired force to joint torques directly. It is able to cope with an arbitrary number of interaction points with the environment, but does not require force measurement at such points and does not use inverse kinematics or dynamics. The controller was tested both in simulation and on a real humanoid [3], and has

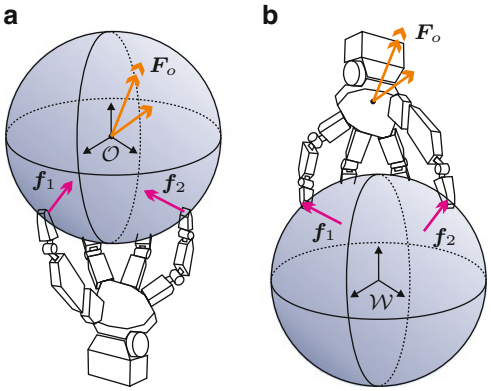



**Fig. 8.2** Force distribution is fundamentally similar in (**a**) Grasping and (**b**) Balancing

been extended to compensate for yaw perturbations and to provide adaptability to unknown rough terrain [8].

More recently, a dynamic balance force controller was proposed for determining full body joint torques based on the desired motion of the Center of Mass (COM), combined with some desired virtual task forces [21]. The approach controls the motion of the COM and the angular momentum of the robot by computing suitable contact forces via a quadratic optimization problem. The mapping of the contact forces to the joint torques is solved considering the nonlinear multi-body dynamics of the system. In addition to the force distribution, the control of internal forces during multicontact interaction tasks was studied in [19], based on the concept of a virtual-linkage model, which provides a representation of the internal and COM resultant forces with respect to reaction forces on the supporting surfaces. A balancing controller based on the independent control of the desired ground reaction force and center of pressure at each support foot was also proposed [11], which allows dealing with different ground geometry. The approach minimizes ankle torques while generating desired rates of change of momenta. The performance of the approach is shown with simulated experiments.

This work presents an overview of a new approach for a posture controller able to deal with unknown external perturbations by distributing the required balancing forces among predefined contact points [16]. The approach is strongly based on the observation that the problems of grasping an object and balancing a robot are fundamentally similar, in the sense that both try to achieve a desired wrench $\boldsymbol{F}_o$ (on the object in the grasping case, on the robot in the balancing case) based on the application of suitable forces at the contact points $\boldsymbol{f}_i$ (at the fingertips or at the feet). Figure 8.2 illustrates such similarity. By using the basic theory of grasping, the force required to counteract external perturbations is distributed to the contact points; the final solution is obtained via a constrained optimization problem. The approach is validated in simulation, and tested on the DLR biped.

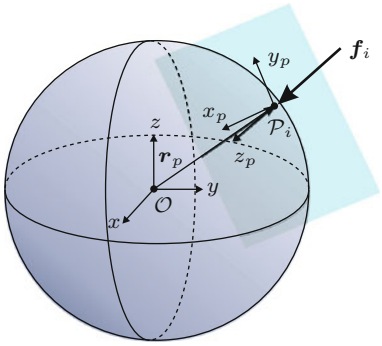



**Fig. 8.3** Object and contact coordinate frames

## 8.2 Grasping Basics

An object is grasped for resisting external perturbations or to manipulate it in a dexterous way. The fulfillment of such tasks depends on the selection of suitable forces $\boldsymbol{f}_i$ applied at contact points $P_i$ such that they produce a desired net wrench $\boldsymbol{F}_o$ on the object [13]. Each contact location is described by its relative position $\boldsymbol{r}_p$ and orientation $\boldsymbol{R}_p$ with respect to the object reference frame $\mathcal{O}$, commonly located at the COM of the object (Fig. 8.3).

In general, a fingertip can only apply forces in certain directions, described by a contact model. The friction properties at the fingertip are commonly described using Coulomb's friction model, which states that slippage is avoided when $f^t \leq \mu f^n$, where $f^n$ and $f^t$ are the magnitudes of the normal and tangential components, respectively, and $\mu$ is the friction coefficient. Therefore, the set of allowable contact forces at the contact point is

$$\mathcal{F}_i = \left\{ \boldsymbol{f}_i \in \mathbb{R}^3 \middle| \sqrt{f_{i_x}^2 + f_{i_y}^2} \leq \mu f_{i_z}, f_{i_z} \geq 0 \right\} \tag{8.1}$$

Geometrically, $\mathcal{F}_i$ represents a *friction cone* with axis along the surface normal and a semiangle of $\varphi = \text{atan}(\mu)$. Besides the friction constraints, the forces must also fulfill the positivity restriction, i.e. the fingers can push but cannot pull the object.

The generalized force $\boldsymbol{F}_i$ that can be applied at a contact point is described by $\boldsymbol{F}_i = \boldsymbol{B}_i \boldsymbol{f}_i$, with $\boldsymbol{B}_i$ being the *wrench basis* that characterizes the contact model. For instance, for a frictional point contact, the applied wrench is

$$\boldsymbol{F}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{i_x} \\ f_{i_y} \\ f_{i_z} \end{bmatrix} = \boldsymbol{B}_i \boldsymbol{f}_i \tag{8.2}$$

The wrench exerted by a single contact on the object, expressed in the object coordinate frame, is given by

$$\boldsymbol{F}_{O_i} = \boldsymbol{Ad}_{op}^T \boldsymbol{B}_i \boldsymbol{f}_i = \boldsymbol{G}_i \boldsymbol{f}_i \tag{8.3}$$

where $\boldsymbol{G}_i = \boldsymbol{Ad}_{op}^T \boldsymbol{B}_i$ is called the contact map. $\boldsymbol{Ad}_{op}^T$ is the transpose of the adjoint matrix for the homogeneous transformation from the frame $\mathcal{P}$ to $\mathcal{O}$, given by

$$\boldsymbol{Ad}_{op}^T = \begin{pmatrix} \boldsymbol{R}_p & 0 \\ \hat{\boldsymbol{r}}_p \boldsymbol{R}_p & \boldsymbol{R}_p \end{pmatrix} \tag{8.4}$$

with $\hat{\boldsymbol{r}}_p$ the cross product matrix for the vector $\boldsymbol{r}_p = \begin{pmatrix} x_p & y_p & z_p \end{pmatrix}^T$, given by

$$\hat{\boldsymbol{r}}_p = \begin{pmatrix} 0 & -z_p & y_p \\ z_p & 0 & -x_p \\ -y_p & x_p & 0 \end{pmatrix} \tag{8.5}$$

The total wrench $\boldsymbol{F}_O$ on the object is the sum of the contributions from each one of the $\eta$ contacts, expressed in the same coordinate frame $\mathcal{O}$,

$$\boldsymbol{F}_O = \begin{bmatrix} \boldsymbol{G}_1 \dots \boldsymbol{G}_\eta \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_\eta \end{bmatrix} = \boldsymbol{G} \boldsymbol{f}_C \tag{8.6}$$

where $\boldsymbol{G}$ is the *grasp map*, given by

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{Ad}_{op_{-1}}^T \boldsymbol{B}_1 \cdots \boldsymbol{Ad}_{op_\eta}^T \boldsymbol{B}_\eta \end{bmatrix} \tag{8.7}$$

Assuming $\eta$ frictional point contacts, the grasp map is further simplified to

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{R}_{p_1} & \cdots & \boldsymbol{R}_{p_\eta} \\ \hat{\boldsymbol{r}}_{p_1} \boldsymbol{R}_{p_1} & \cdots & \hat{\boldsymbol{r}}_{p_\eta} \boldsymbol{R}_{p_\eta} \end{pmatrix} \tag{8.8}$$

## 8.3 Dynamic Model

The unconstrained dynamics of a biped robot can be described using the velocity vector $\boldsymbol{v} = (\boldsymbol{\nu}^T, \boldsymbol{\omega}^T, \dot{\boldsymbol{q}}^T)^T$, where $\boldsymbol{\nu} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ are the linear and angular velocities of a frame $\mathcal{B}$ attached to a base link (e.g. trunk) with respect to the world coordinate frame $\mathcal{W}$, and $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the angular velocities for the $n$ joints of the

robot. Let $\boldsymbol{r}_b \in \mathbb{R}^3$ and $\boldsymbol{R}_b \in SO(3)$ be the position and orientation of the base frame $\mathcal{B}$; then, the dynamical model is

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \boldsymbol{p}(\boldsymbol{q}, \boldsymbol{R}_b) = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{\tau} \end{pmatrix} + \sum_{k=\{r,l\}} \boldsymbol{J}_k(\boldsymbol{q})^T \boldsymbol{F}_k \tag{8.9}$$

where $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v}$, and $\boldsymbol{p}(\boldsymbol{q}, \boldsymbol{R}_b)$ are the robot's inertia matrix, the vector of centrifugal and Coriolis terms, and the vector of gravity terms, respectively; $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuator torques, and $\boldsymbol{F}_k \in \mathbb{R}^6$ are the body wrenches acting at the robot's right ($k = r$) and left foot ($k = l$). Moreover, $\boldsymbol{J}_k(\boldsymbol{q}) = [\boldsymbol{Ad}_{kb}(\boldsymbol{q})\ \boldsymbol{J}_k^b(\boldsymbol{q})]$, with $\boldsymbol{Ad}_{kb}(\boldsymbol{q})$ being the adjoint matrix for the homogeneous transformation between the feet and the base link, and $\boldsymbol{J}_k^b(\boldsymbol{q})$ is the body Jacobian for the feet [13].

This dynamical model is further simplified if the COM velocity $\dot{\boldsymbol{r}}_C$ is used instead of the velocity of the base link [23]. Then, $\boldsymbol{v}$ is replaced by $\boldsymbol{v}_C = (\dot{\boldsymbol{r}}_C^T, \boldsymbol{\omega}^T, \dot{\boldsymbol{q}}^T)^T$ via

$$\boldsymbol{v} = \begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{R}_b^T & {}^B\hat{\boldsymbol{r}}_C(\boldsymbol{q}) & -\boldsymbol{J}_{BC}(\boldsymbol{q}) \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{pmatrix} \dot{\boldsymbol{r}}_C \\ \boldsymbol{\omega} \\ \dot{\boldsymbol{q}} \end{pmatrix}}_{\boldsymbol{v}_C} \tag{8.10}$$

where $\boldsymbol{J}_{BC}(\boldsymbol{q}) = \partial {}^B\boldsymbol{r}_C(\boldsymbol{q})/\partial \boldsymbol{q}$, and ${}^B\boldsymbol{r}_C(\boldsymbol{q})$ is the COM position represented in the coordinate frame $\mathcal{B}$. With these new coordinates, the dynamic model is transformed to

$$\begin{bmatrix} m\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{M}}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{v}}_C + \begin{pmatrix} \boldsymbol{0} \\ \bar{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{v}_C)\boldsymbol{v}_C \end{pmatrix} + \begin{pmatrix} m\boldsymbol{g} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \underline{\boldsymbol{0}} \\ \boldsymbol{0} \\ \boldsymbol{\tau} \end{pmatrix} + \sum_{k=\{r,l\}} \bar{\boldsymbol{J}}_k(\boldsymbol{q})^T \boldsymbol{F}_k \tag{8.11}$$

where $m$ is the total mass, and $\bar{\boldsymbol{M}}(\boldsymbol{q})$ and $\bar{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{v}_C)\boldsymbol{v}_C$ are terms resulting from the coordinate transformation (8.10). The resulting Jacobian matrices are given by $\bar{\boldsymbol{J}}_k(\boldsymbol{q}) = \boldsymbol{J}_k(\boldsymbol{q})\boldsymbol{A}$, with $k = \{r, l\}$, and can be partitioned as

$$\bar{\boldsymbol{J}}_k(\boldsymbol{q}) = \left[ \begin{pmatrix} \boldsymbol{R}_k^T \\ \boldsymbol{0} \end{pmatrix} \quad \boldsymbol{Ad}_{kb} \begin{pmatrix} {}^B\hat{\boldsymbol{r}}_C \\ \boldsymbol{I} \end{pmatrix} \quad \underbrace{\boldsymbol{J}_k^b(\boldsymbol{q}) - \boldsymbol{R}_{kb}\boldsymbol{J}_{BC}(\boldsymbol{q})}_{\boldsymbol{J}_{ck}(\boldsymbol{q})} \right] \tag{8.12}$$

The COM dynamics emerges from the first three equations in (8.11), i.e.

$$m\ddot{\boldsymbol{r}}_C + m\boldsymbol{g} = \sum_{k=r,l} \boldsymbol{R}_k \boldsymbol{f}_k \tag{8.13}$$

and it is influenced by the force components $\boldsymbol{f}_k$ from the contact wrenches $\boldsymbol{F}_k = (\boldsymbol{f}_k, \boldsymbol{t}_k)$. The last $n$ equations in (8.11) provide a kinetostatic mapping of the contact wrenches $\boldsymbol{F}_r$, $\boldsymbol{F}_l$ at the right and left foot to the joint torques $\boldsymbol{\tau}$ via

$$\boldsymbol{\tau} = \boldsymbol{J}_{cr}(\boldsymbol{q})^T \boldsymbol{F}_r + \boldsymbol{J}_{cl}(\boldsymbol{q})^T \boldsymbol{F}_l \tag{8.14}$$

The representation provided in (8.11), which gives the interaction of the isolated COM dynamics with the remaining multibody dynamics, was previously used for the design of a force based COM balancing controller [9]. In the following section, we extend the controller from [9] by adding a posture controller for the base orientation, and using the force distribution framework described in Sect. 8.2.

## 8.4 Balancing Controller

This section presents a balancing controller which regulates the position of the robot's total COM in $\mathcal{W}$, $\boldsymbol{r}_C \in \mathbb{R}^3$, and the orientation $\boldsymbol{R}_b$ of the base link. Figure 8.4 shows the structure of the proposed controller. Basically, given a desired equilibrium position $\boldsymbol{r}_C^d$ for the COM position and a desired orientation $\boldsymbol{R}_d$ of the trunk, we compute a desired wrench $\boldsymbol{F}_{GA}$ to be applied to the robot. That wrench is distributed to forces $\boldsymbol{f}_C \in \mathbb{R}^{3\eta}$ at the $\eta$ supporting contact points at the feet. In order to realize the desired contact forces, we compute the resulting contact wrenches $\boldsymbol{F}_r$ and $\boldsymbol{F}_l$ at the right and left foot, and map these contact wrenches to corresponding joint torques $\boldsymbol{\tau}_d$ via (8.14), thus avoiding the use of inverse kinematics or dynamics. The resulting joint torques are commanded to an underlying joint torque controller [1]. Further details are provided below.

### *8.4.1 Object Force Generation*

A biped robot with multiple contacts with the ground can be analyzed as a series of contact forces applied at the contact points, which generate a net wrench on the robot according to (8.6). The force component of the net wrench is called the ground reaction force $\boldsymbol{f}_{GR}$. For the balancing controller, it is more convenient to consider a desired wrench that must be applied at the COM of the robot, which must be generated through forces applied at the contact points. The net force that must be generated is called ground applied force $\boldsymbol{f}_{GA}$, defined as $\boldsymbol{f}_{GA} = -\boldsymbol{f}_{GR}$.

To keep the desired COM position, we obtain a desired ground applied force $\boldsymbol{f}_{GA}^d$, given by the task of recovering the initial position (according to a compliance control law) while providing gravity compensation, i.e.

$$\boldsymbol{f}_{GA}^d = m(\boldsymbol{g} + \ddot{\boldsymbol{r}}_C^d) - \boldsymbol{K}_P(\boldsymbol{r}_C - \boldsymbol{r}_C^d) - \boldsymbol{D}_P(\dot{\boldsymbol{r}}_C - \dot{\boldsymbol{r}}_C^d) \tag{8.15}$$

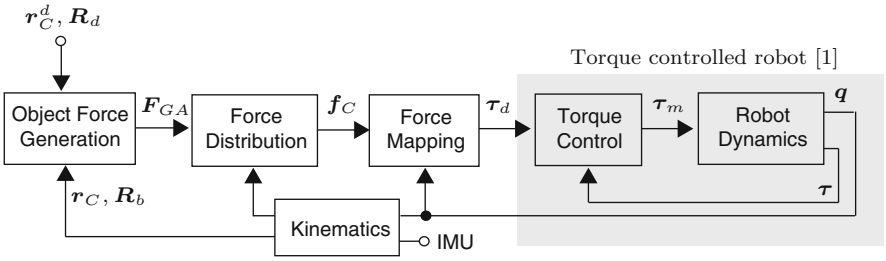



**Fig. 8.4** Overview of the balancing controller

where $m$ is the total mass of the robot, $\boldsymbol{g}$ is the gravity vector, $\boldsymbol{K}_P, \boldsymbol{D}_P > 0$ are the proportional and differential gain matrices, and $\boldsymbol{r}_C^d$, $\dot{\boldsymbol{r}}_C^d$, and $\ddot{\boldsymbol{r}}_C^d$ are the desired position, velocity and acceleration of the COM.

To deal with the orientation perturbations, let $\delta$ and $\boldsymbol{\epsilon}$ be the scalar and vector part of the quaternion representation of $\boldsymbol{R}_{db} = \boldsymbol{R}_d^T \boldsymbol{R}_b$, with $\boldsymbol{R}_b$ and $\boldsymbol{R}_d$ the current and desired trunk orientation. The required torque for a torsional spring acting to align $\boldsymbol{R}_b$ to $\boldsymbol{R}_d$ can be derived from a potential function $V_O = 2\boldsymbol{\epsilon}^T \boldsymbol{K}_r \boldsymbol{\epsilon}$, and is given by $\boldsymbol{\tau}_r = -2(\delta \boldsymbol{I} + \hat{\boldsymbol{\epsilon}})\boldsymbol{K}_r \boldsymbol{\epsilon}$, with $\boldsymbol{K}_r \in \mathbb{R}^{3\times3}$ a symmetric and positive definite stiffness matrix. This relation can be verified by using the principle of virtual work [2]. Thus, a PD-like orientation controller for the trunk is given by

$$\boldsymbol{\tau}_{GA}^d = \boldsymbol{R}_{wb}(\boldsymbol{\tau}_r - \boldsymbol{D}_r(\boldsymbol{\omega} - \boldsymbol{\omega}^d)) \tag{8.16}$$

with $\boldsymbol{D}_r \in \mathbb{R}^{3\times3}$ being a symmetric and positive definite damping matrix.

Finally, the desired force and torque from (8.15) and (8.16) are combined to get the net desired wrench $\boldsymbol{F}_{GA} = (\boldsymbol{f}_{GA}^d, \boldsymbol{\tau}_{GA}^d)^T$.

### 8.4.2 Contact Force Distribution

Consider a biped robot with $\eta$ contact points with the ground (Fig. 8.5), and let $\boldsymbol{r}_i = [x_i, y_i, z_i]^T$ with $i = 1, ..., \eta$ be the position of the contact points with respect to the COM (i.e. in the coordinate frame $\mathcal{O}$). The robot only interacts with the environment through forces $\boldsymbol{f}_i$ at the $\eta$ frictional contact points. Each contact force is described as $\boldsymbol{f}_i = [f_{i_x}, f_{i_y}, f_{i_z}]^T$, and all the contact forces are stacked in the contact force vector $\boldsymbol{f}_C \in \mathbb{R}^{3\eta}$.

The desired net wrench on the robot, $\boldsymbol{F}_{GA}$, must be generated through suitable contact forces $\boldsymbol{f}_C$; the relation between them is given by (8.6) ($\boldsymbol{F}_{GA} = \boldsymbol{G}_C \boldsymbol{f}_C$), with $\boldsymbol{G}_C$ the *contact map* (i.e. the grasp map applied to walking robots). For instance, for the case of a robot standing on flat ground, the coordinate frame at each contact point can be chosen parallel to the world frame $\mathcal{W}$, and $\boldsymbol{G}_C \in \mathbb{R}^{6\times3\eta}$ gets the simplified form

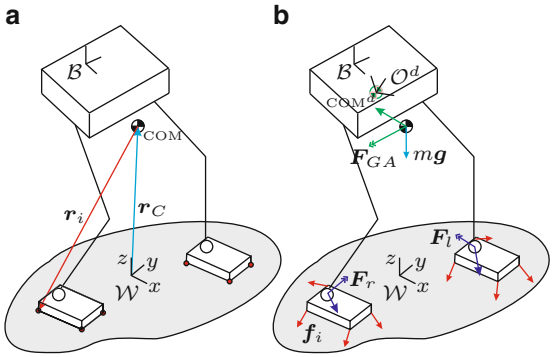



**Fig. 8.5** Contact positions and forces for a biped robot: (**a**) Location of the COM; (**b**) Forces

$$\boldsymbol{G}_C = \begin{pmatrix} \boldsymbol{I}_{3\times 3} & \cdots & \boldsymbol{I}_{3\times 3} \\ \hat{\boldsymbol{r}}_{p_1} & \cdots & \hat{\boldsymbol{r}}_{p_\eta} \end{pmatrix} \tag{8.17}$$

For a given $\boldsymbol{F}_{GA}$, the corresponding contact forces can be computed as

$$\boldsymbol{f}_C = \boldsymbol{G}_C^{\#} \boldsymbol{F}_{GA} \tag{8.18}$$

with $\boldsymbol{G}_C^{\#} = \boldsymbol{G}_C^T (\boldsymbol{G}_C \boldsymbol{G}_C^T)^{-1}$ the pseudoinverse of the contact map. This solution minimizes the Euclidean norm of the contact forces under the constraint (8.6). The solution to this problem has also been considered in the grasping community as the minimization of the grasping forces that ensure a stable grasp [6, 18]. These optimization procedures include the friction cone restrictions, and guarantee that the equilibrium (8.6) is always fulfilled. However, in the case of the balancing problem the wrench $\boldsymbol{F}_{GA}$ on the object is a control command resulting from (8.15) and (8.16), which might not be exactly met. In order to ensure that all constraints are fulfilled, a different approach is required for this optimization problem.

### 8.4.3 Force Distribution Using Unilateral Constraints

To get a desired net wrench $\boldsymbol{F}_{GA}$, the distribution of contact forces according to (8.18) in general does not guarantee that the positivity restriction and the friction constraints (8.1) at the contact points are fulfilled. As an alternative, the contact forces can be computed with a constrained multi-objective optimization problem. The main objective is to achieve a desired force on the COM, which can be formulated as the minimization of the cost function $J_1(\boldsymbol{f}_C) = ||[\boldsymbol{I}\ \boldsymbol{0}](\boldsymbol{F}_{GA} - \boldsymbol{G}_C \boldsymbol{f}_C)||_2^2$. Getting the object torque is a secondary objective, with an objective function $J_2(\boldsymbol{f}_C) = ||[\boldsymbol{0}\ \boldsymbol{I}](\boldsymbol{F}_{GA} - \boldsymbol{G}_C \boldsymbol{f}_C)||_2^2$. Additionally, a third objective is the minimization of the Euclidean norm of the contact forces, i.e. $J_3(\boldsymbol{f}_C) = \boldsymbol{f}_C^T \boldsymbol{f}_C$. The three objectives are combined in a single objective function $J(\boldsymbol{f}_C)$

$$J(\boldsymbol{f}_C) = \alpha_1 J_1(\boldsymbol{f}_C) + \alpha_2 J_2(\boldsymbol{f}_C) + \alpha_3 J_3(\boldsymbol{f}_C) \tag{8.19}$$

where $\alpha_1, \alpha_2, \alpha_3 > 0$ are the corresponding weights for the three objectives. By choosing $\alpha_3 << \alpha_2 << \alpha_1$, the first objective is selected as the main priority task and the third objective acts mainly as a regularization of the Hessian for the objective function. The minimization of $J(\boldsymbol{f}_C)$ subject to restrictions arising from a polyhedral approximation to the friction cone (8.1) and from the positivity constraints represents a quadratic optimization problem, given by

$$min_{f_C} \ \boldsymbol{f}_C^T \boldsymbol{Q} \boldsymbol{f}_C + \boldsymbol{p}^T \boldsymbol{f}_C \tag{8.20}$$

where

$$\boldsymbol{Q} = \alpha_3 \boldsymbol{I} + \boldsymbol{G}_C^T \begin{pmatrix} \alpha_1 \boldsymbol{I} & 0 \\ 0 & \alpha_2 \boldsymbol{I} \end{pmatrix} \boldsymbol{G}_C \tag{8.21}$$

$$\boldsymbol{p} = -\boldsymbol{G}_C^T \begin{pmatrix} \alpha_1 \boldsymbol{I} & 0 \\ 0 & \alpha_2 \boldsymbol{I} \end{pmatrix} \boldsymbol{F}_{GA} \tag{8.22}$$

Note that the positivity and friction constraints force the ZMP to lie within the support polygon. Based on (8.14), additional restrictions such as torque limits at each joint can also be included in the problem statement.

The proposed approach assumes that all the predefined contacts are active at every moment, and therefore distributes the desired wrench to all the contact points. To guarantee that each contact is always active on the real robot, a lower limit for the contact force can be preset to some positive value.

## 8.5 Experiments

The balancing controller was tested in simulations using OpenHRP3 [10], and in experiments with the DLR Biped [15] (Fig. 8.1). The DLR Biped has six degrees of freedom per leg, a 6-DOF force-torque sensor (FTS) in each foot, position and torque sensors at each joint, and an inertial measurement unit (IMU). The FTS information is not used, since the proposed control scheme does not require any measurement of the contact forces at the feet.

### *8.5.1 Implementation Details*

The origin of the world coordinate frame $\mathcal{W}$ was chosen to be in the middle point between the two feet. The trunk orientation and angular velocity are measured via

the IMU. As the IMU shows considerable drift in the yaw rotation, we approximate the yaw angle between the world frame and the trunk by comparing the trunk orientation to the baseline between the right and left foot.

The proportional gain matrix $\boldsymbol{K}_p$ for the controller was chosen as a diagonal matrix with stiffness values of $k_h = 900\ N/m$ in the horizontal ($x$ and $y$) directions, and $k_v = 3{,}000\ N/m$ in the vertical ($z$) direction. The damping gain matrix $\boldsymbol{D}_p$ was chosen as a diagonal matrix with the elements set to $d_h = \sqrt{(mk_h)}2 \cdot 0.8$ for the horizontal components, and $d_v = \sqrt{(mk_v)}2 \cdot 0.2$ for the vertical component. The rotational stiffness and damping matrices were set to $\boldsymbol{K}_r = 100\boldsymbol{I}[Nm/rad]$ and $\boldsymbol{D}_r = 50\boldsymbol{I}[Nms/rad]$. The same controller gains were used in the simulations and in the experiments.

The constrained optimization problem from Sect. 8.4.3 was solved using the open source software qpOASES [5]. The values of the weights for the multi-objective optimization are chosen as $\alpha_1 = 1$, $\alpha_2 = 10^{-3}$ and $\alpha_3 = 10^{-6}$. With $\eta = 8$ frictional contact points, the computation time for the optimization of $3\eta = 24$ components in $\boldsymbol{f}_C$ took less than 200 $\mu s$ on the onboard 2.8 $GHz$ mobile CPU running under the real-time operating system VxWorks.

### 8.5.2 Simulation

As an example of the behavior of the controller, a simulation of an external perturbation was performed. The desired COM velocity is set to $\dot{\boldsymbol{r}}_C^d = 0$, and a lateral force of 70 $N$ was applied during 50 $ms$ at the hip (Fig. 8.6). This force applies also a torque on the base link, since it is not exerted at the COM. Figure 8.7a shows the COM error resulting from this simulation. The components of the contact forces are displayed in Fig. 8.7b. At $t \approx 0.25s$ the vertical forces at the right foot (in blue) reach their lower limit of 4 $N$ set in the optimization algorithm from Sect. 8.4.3.

### 8.5.3 Experimental Evaluation

An experiment of physical interaction with a human is shown in Fig. 8.8, where the trunk of the robot is pushed to create different perturbations in position and orientation. Figure 8.9a shows the corresponding COM error for pushes in $x$ and $y$. Figure 8.9b shows the corresponding contact forces generated by the balancing controller. In this example, the non-negativity constraints on the vertical force components act over a longer duration compared to the simulated experiment, as the human interaction acts as a low frequency disturbance.

Finally, note that in the derivation of the controller there was no assumption for the robot to be on flat ground. When the global trunk orientation can be obtained from an onboard IMU, the controller can deal with uneven terrain. Figure 8.10

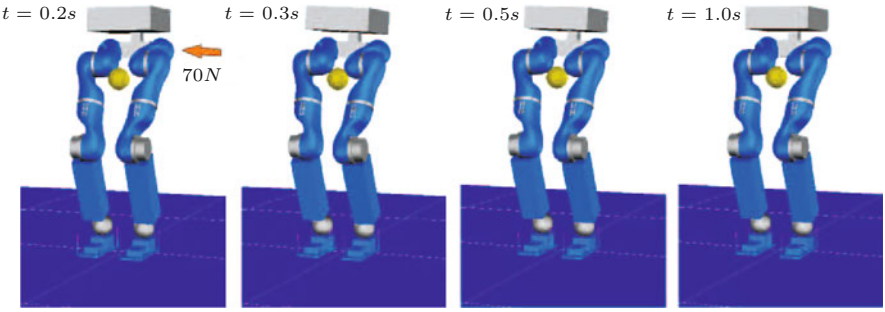



**Fig. 8.6** Balancing experiment in OpenHRP

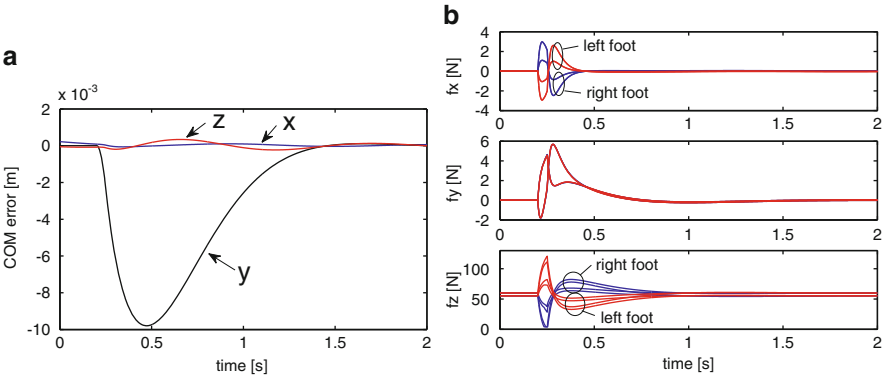



**Fig. 8.7** Results for the simulated lateral force disturbance: (**a**) COM error in $x$ (*blue*), $y$ (*black*), and $z$ (*red*); (**b**) Contact force components at the *right* (*blue*) and *left* (*red*) feet

presents a time sequence of the behavior of the robot mounted on top of a balancing board, and subjected to perturbations coming from a human.

## 8.6 Summary

This chapter has presented an optimization based balancing algorithm for biped robots, which regulates the position of the COM and the orientation of the trunk. It allows to distribute a net wrench, required to recover a desired posture, onto a predefined set of contact points. The approach was verified in simulation and in experiments with a torque controlled robot. The balancing controller, including the approach for distributing the contact force, is general enough to be applied to a biped robot in different contact situations, such as single or double support phases.

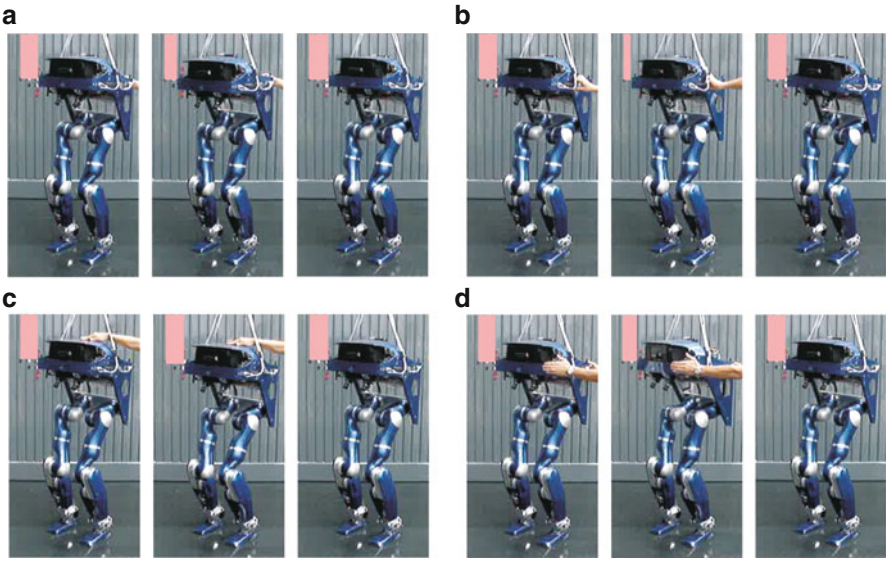



**Fig. 8.8** Compensatory motions for different perturbations applied to the robot: (**a**) in *x*; (**b**) in *y*; (**c**) in *z*; (**d**) in yaw. Filled rectangles are overimposed as an aid to perceive the displacement of the robot

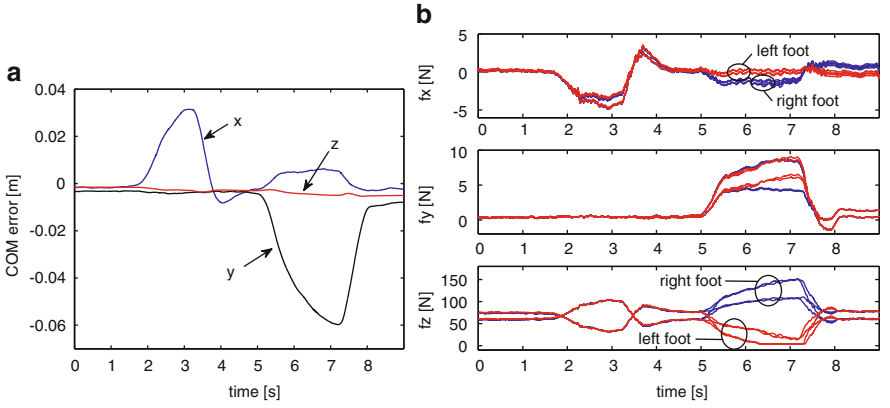



**Fig. 8.9** Results for the experiment of physical human interaction: (**a**) COM error in *x* (*blue*), *y* (*black*), and *z* (*red*); (**b**) Contact force components at the *right* (*blue*) and *left* (*red*) feet

It can also be applied to multi-legged robots or to multi-point contact situations, for instance, to balance forces created when a humanoid robot manipulates an object or interacts with the environment.

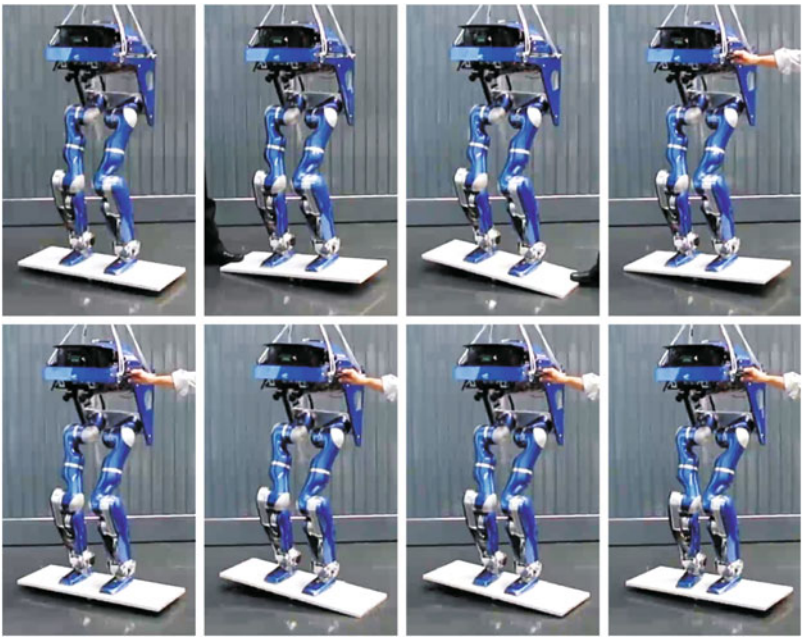


**Fig. 8.10** Behavior of the robot mounted on top of a balancing board

**Acknowledgement** This research is partly supported by the Initiative and Networking Fund of the Helmholtz Association (VH-NG-808).

## References

1. Albu-Schäffer A, Ott C, Hirzinger G (2007) A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. Int J Robot Res **26**(1):23–39
2. Caccavale F, Natale C, Siciliano B, Villani L (1999) Six-dof impedance control based on angle/axis representations. IEEE Trans Robot Autom **15**(2):289–299
3. Cheng G, Hyon SH, Morimoto J, Ude A, Hale JG, Colvin G, Scroggin W, Jacobsen SC (2007) CB: a humanoid research platform for exploring neuroscience. Adv Robotics **21** (10):1097–1114
4. Choui Y, Kim D, You BJ (2006) On the walking control for humanoid robot based on the kinematic resolution of CoM jacobian with embedded motion. In: IEEE international conference on robotics and automation, pp 2655–2660
5. Ferreau H, Bock H, Diehl M (2008) An online active set strategy to overcome the limitations of explicit MPC. Int J Robust Nonlinear Control **18**(8):816–830
6. Han L, Trinkle J, Li Z (2000) Grasp analysis as linear matrix inequality problems. IEEE Trans Robot Autom **16**(6):663–674
7. Hirai K, Hirose M, Haikawa Y, Takenaka T (1998) The development of the Honda humanoid robot. In: IEEE international conference on robotics and automation, pp 1321–1326
8. Hyon SH (2009) Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. IEEE Trans Robot **25**(1):171–178

9. Hyon SH, Hale JG, Cheng G (2007) Full-body compliant human-humanoid interaction: balancing in the presence of unknown external forces. IEEE Trans Robotics **23**(5):884–898
10. Kanehiro F, Fujiwara K, Kajita S, Yokoi K, Kaneko K, Hirukawa H, Nakamura Y, Yamane K (2002) Open architecture humanoid robotics platform. In: IEEE international conference on robotics and automation, pp 24–30
11. Lee SH, Goswami A (2010) Ground reaction force control at each foot: a momentum-based humanoid balance controller for non-level and non-stationary ground. In: IEEE/RSJ international conference on intelligent robots and systems, pp 3157–3162
12. Macchietto A, Zordan V, Shelton C (2009) Momentum control for balance. ACM Trans Graphics **28**(3)
13. Murray R, Li Z, Sastry S (1994) A mathematical introduction to robotic manipulation. CRC, Boca Raton
14. Ott C, Albu-Schäffer A, Kugi A, Hirzinger G (2008) On the passivity based impedance control of flexible joint robots. IEEE Trans Robotics **24**(2):416–429
15. Ott C, Baumgärtner C, Mayr J, Fuchs M, Burger R, Lee D, Eiberger O, Albu-Schäffer A, Grebenstein M, Hirzinger G (2010) Development of a biped robot with torque controlled joints. In: IEEE-RAS international conference on humanoid robots, pp 167–173
16. Ott C, Roa M, Hirzinger G (2011) Posture and balance control for biped robots based on contact force optimization. In: IEEE-RAS international conference on humanoid robots, pp 26–33
17. Pratt J, Krupp B (2008) Design of a bipedal walking robot. In: Proceedings of the 2008 SPIE, vol 6962
18. Saut J, Remond C, Perdereau V, Drouin M (2005) Online computation of grasping force in multi-fingered hands. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1223–1228
19. Sentis L, Park J, Khatib O (2010) Compliant control of multicontact and center-of-mass behaviors in humanoid robots. IEEE Trans Robotics **26**(3):483–501
20. Setiawan S, Hyon S, Yamaguchi J, Takanishi A (1999) Physical interaction between human and a bipedal humanoid robot - realization of human-follow walking. In: IEEE international conference on robotics and automation, pp 361–367
21. Stephens B, Atkeson CG (2010) Dynamic balance force control for compliant humanoid robots. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1248–1255
22. Sugihara T, Nakamura Y (2002) Whole-body cooperative balancing of humanoid robot using COG jacobian. In: IEEE/RSJ international conference on intelligent robots and systems, pp 2575–2580
23. Wieber PB (2006) Holonomy and nonholonomy in the dynamics of articulated motion. In: Fast motions in biomechanics and robotics, vol 340, Lecture notes in control and information sciences. Springer, Berlin/Heidelberg, pp 411–425

# Chapter 9
# Robot-Based Testing of Total Joint Replacements

**Christoph Woernle, Michael Kähler, Roman Rachholz, János Zierath, Sven Herrmann, Robert Souffrant, Daniel Kluess, and Rainer Bader**

**Abstract** Instabilities of artificial joints are prevalent complications in total joint arthroplasty. In order to investigate failure mechanisms like dislocation of total hip replacements or instability of total knee replacements, a novel test approach is introduced by means of a hardware-in-the-loop (HiL) simulation combining the advantages of an experimental with a numerical approach. The HiL simulation is based on a six-axes industrial robot and a musculoskeletal multibody model. Within the multibody model, the anatomical environment of the correspondent joint is represented such that the soft tissue response is considered during an instability event. Hence, the robot loads and moves the real implant components according to the data provided by the multibody model while transferring back the relative displacement of the implant components and the resisting moments recorded. HiL simulations provide a new biomechanical testing tool which enables comparable and reproducible investigations of various joint replacement systems with respect to their instability behaviour under realistic movements and physiological load conditions.

C. Woernle (✉) • M. Kähler • R. Rachholz
Department of Mechanical Engineering and Marine Technology,
University of Rostock, Rostock, Germany
e-mail: woernle@uni-rostock.de; michael.kaehler@uni-rostock.de; roman.rachholz@uni-rostock.de

J. Zierath
W2E Wind-to-Energy GmbH, Rostock, Germany
e-mail: JZierath@wind-to-energy.de

S. Herrmann • R. Souffrant • D. Kluess • R. Bader
Department of Orthopedics, University Medicine Rostock, Rostock, Germany
e-mail: sven.herrmann@med.uni-rostock.de; robert.souffrant@med.uni-rostock.de;
daniel.kluess@med.uni-rostock.de; rainer.bader@med.uni-rostock.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_9, 

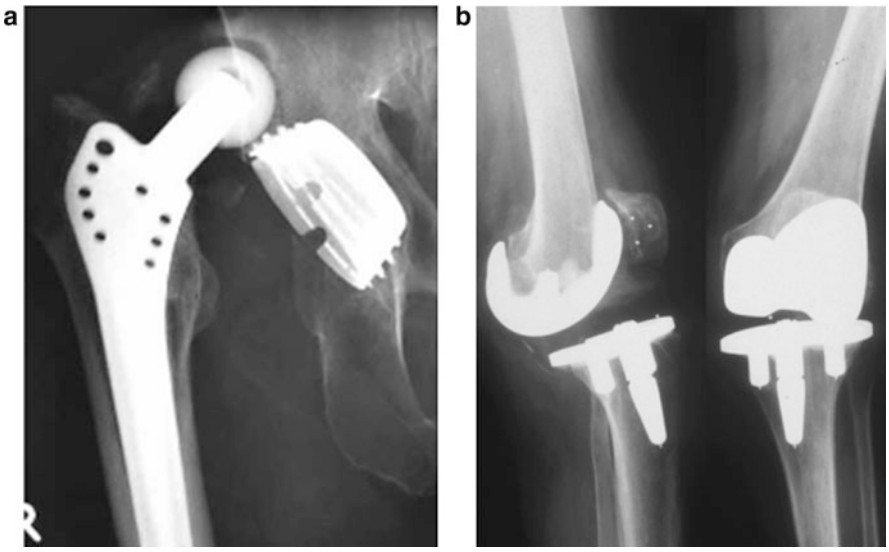



**Fig. 9.1** Failures of total joint replacements. (**a**) Dislocation of a total hip replacement (THR). (**b**) Unstable total knee replacement (TKR)

## 9.1 Introduction

Instability of artificial joints is still one of the most prevalent reasons for revision surgery (Fig. 9.1). With regard to total hip replacements (THR), dislocation of the femoral head represents a major reason for revision procedures [3]. Mechanisms linked to the dislocation process involve prior prosthetic or bony contact (impingement), and spontaneous separation due to dynamic forces. Similarly, instability and adverse kinematics in total knee replacements (TKR) has been reported as one of the most important reasons for implant failure [2]. As TKRs are by design less constrained than THRs, relative movement between joint partners is governed by restraining ligament and muscular forces. Therefore, instability mechanisms are given by excessive relative movement between the femoral and tibial component due to insufficient implant position and surrounding soft tissue and hence resulting in unstable articulation.

Clinical and biomechanical studies addressed the issue of THR and TKR instability highlighting several influencing factors. Soft tissue condition, implant position, and implant design have each been frequently regarded as major factors. However, clinical studies make it difficult to study causes and effects systematically as many factors cannot be kept constant. While in vivo measurements of instability-associated manoeuvres are discouraged by ethical and technical reasons, cadaver studies make it difficult to systematically evaluate influencing factors on joint instability under reproducible boundary conditions. Moreover, numerical studies contain numerous simplification and idealisations concerning friction, material behaviour, and boundary conditions.

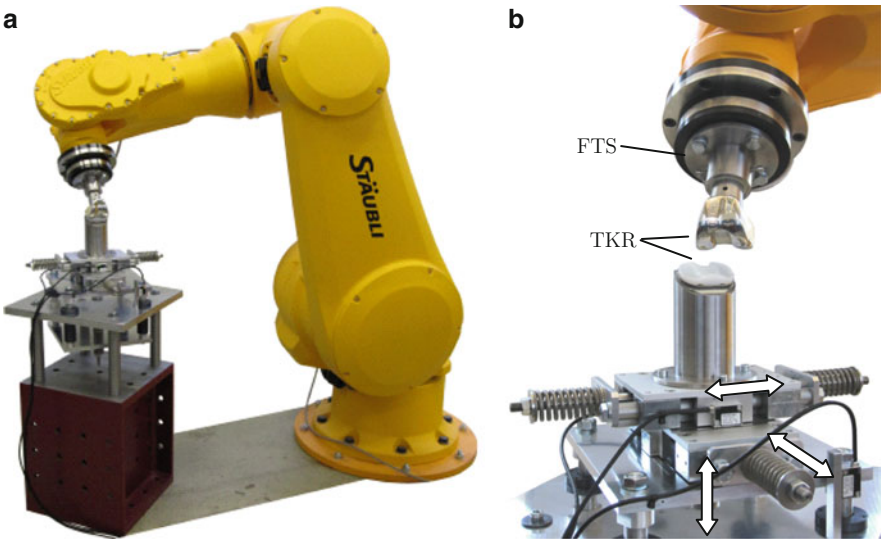



**Fig. 9.2** Physical test setup. (**a**) Industrial robot with a TKR. (**b**) Robot endeffector with femoral component of the TKR and 6D force-torque sensor (FTS) and elastically compliant support with tibial component of the TKR and position sensors

In order to overcome the limitations of experimental and numerical instability analyses, both approaches are combined in a hardware-in-the-loop (HiL) simulation [4, 5, 7, 11]. It is based on a highly flexible mechatronic test system consisting of a musculoskeletal multibody model and an industrial robot as actuator system. In this work, the fundamental concept of the HiL joint simulator is described both for THRs and TKRs. The major goal is to enable comparable investigations of different THR and TKR designs with respect to instability under reproducible, physiological-like boundary conditions which accounts for the soft tissue response during instability scenarios.

## 9.2 Physical Test Setup

A six-axes industrial robot (TX 200, Stäubli Tec-Systems GmbH, Bayreuth, Germany) equipped with a six-dimensional force-torque sensor (ATI Industrial Automation, Apex, NC, USA) is used as actuator system of the physical setup (Fig. 9.2). The robot is capable of generating both the range of motion and the reaction forces necessary for THR and TKR loading. The implant components are attached to the endeffector and an elastically compliant support, respectively, by specifically designed mounting devices. The elastic support is used for force/torque control of the robot. It consists of three serially arranged prismatic joints with orthogonal axes and springs restraining the displacements along the axes and providing compliance in the three translational directions. Likewise, rotational compliance is simultaneously achieved by the translational stiffness of the support.

Force/torque control is realised by outer force regulating control loops that generate the control input for the inner position and velocity controllers. This control method requires a physical compliance somewhere in the kinematical chain closed by the environmental contact of the robot which is ensured by the compliant support. The force regulating control loops enable the robot movement until the actual forces and torques, respectively, measured by the six-dimensional force-torque sensor coincide with the corresponding desired values [9]. Position control is provided by the robot controller (CS8C HP, Stäubli Tec-Systems GmbH, Bayreuth, Germany) running with a control cycle of 4 ms in its standard configuration. It basically consists of cascaded position/velocity feedback loops complemented by a feed-forward inverse dynamics controller.

The displacements along the prismatic axes of the compliant support are measured by three displacement sensors (MSK 5,000, SIKO, Buchenbach, Germany) attached to the framework. Thus, the relative displacements between the implant components can be recorded by comparing the position of the endeffector with respect to the compliant support.

## 9.3 Functional Principle of the HiL Joint Simulator

For HiL simulation of total joint replacements, the robot interacts with a simulation computer on which a biomechanical multibody model is running. The functional principle of the HiL concept is based on complementary sets of free and constrained directions of the artificial joint under investigation as described in the following for THRs and TKRs.

### *9.3.1 Principle of HiL Testing of a THR*

Considering the spatial load case for an artificial hip joint according to Fig. 9.3, the three rotations of the femur relative to the pelvis with rotation angles summarised in vector $\boldsymbol{\beta} \in \mathbb{R}^3$ are free within the anatomical range of motion. The three translational movements of the femoral head center with coordinates summarised in vector $\boldsymbol{z} \in \mathbb{R}^3$ are treated as constrained directions. For an actual time instant $t$, the multibody model delivers values of the rotation angles $\boldsymbol{\beta}$ in the free directions and of the reaction forces in the constrained directions, summarised in the vector $\boldsymbol{f}^{\mathrm{r}} \in \mathbb{R}^3$. Soft tissue forces $\boldsymbol{f}^{\mathrm{s}}$ comprising passive ligament and capsule forces and active muscle forces as well as gravitational and inertial forces are also taken into account. Both rotation angles $\boldsymbol{\beta}$ and reaction forces $\boldsymbol{f}^{\mathrm{r}}$ are transferred to the robot controller. Hence, the robot rotates the femoral component into the position $\bar{\boldsymbol{\beta}}$ and applies the reaction forces $\bar{\boldsymbol{f}}^{\mathrm{r}}$ onto the endoprosthesis. In the case of ideal transmission of the magnitudes, the robot values $\bar{\boldsymbol{\beta}}$ and $\bar{\boldsymbol{f}}^{\mathrm{r}}$ are identical with the corresponding values $\boldsymbol{\beta}$ and $\boldsymbol{f}^{\mathrm{r}}$ of the model. However, in reality differences occur

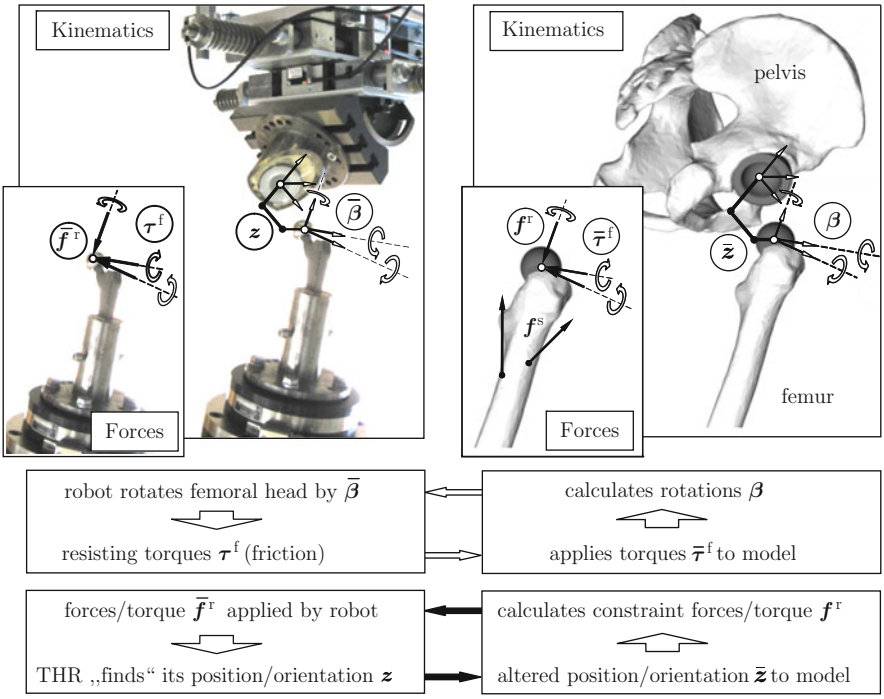



**Fig. 9.3** HiL simulation for testing THRs with illustration of the exchanged values between the musculoskeletal multibody model (*right*) and the actuator system (*left*) at the kinematic and force levels

due to limited dynamic bandwidth of the controlled robot, signal delay times, and other errors.

To close the HiL control loop, the displacement components of the femoral head center $\boldsymbol{z} \in \mathbb{R}^3$ are measured and fed back to the multibody model. Another loop is closed by the resisting forces in the unconstrained directions, here the torque components $\boldsymbol{\tau}^{\mathrm{f}} \in \mathbb{R}^3$ along the coordinates $\bar{\boldsymbol{\beta}}$, which are also measured and fed back to the multibody model. The resisting torques $\boldsymbol{\tau}^{\mathrm{f}}$ could be caused by friction forces in the THR. Again disturbances of the measurements and the signal transmissions cause differences between the actual robot values $\boldsymbol{\tau}^{\mathrm{f}}$ and $\boldsymbol{z}$ and the corresponding transferred values $\bar{\boldsymbol{\tau}}^{\mathrm{f}}$ and $\bar{\boldsymbol{z}}$.

The robot is able to apply the reaction force components $\bar{\boldsymbol{f}}^{\mathrm{r}}$ if the endoprosthesis withstands these loads in the corresponding directions. Then, no relative translational displacement occurs, thus $\boldsymbol{z} = \boldsymbol{0}$. Otherwise, the femoral head is moved out of the acetabular cup by a displacement $\boldsymbol{z}$ indicating a dislocation event.

### *9.3.2 Principle of HiL Testing of a TKR*

For HiL testing of artificial knee joints according to Fig. 9.4, the same testing principle with different free and constrained directions is applied. Here, the free

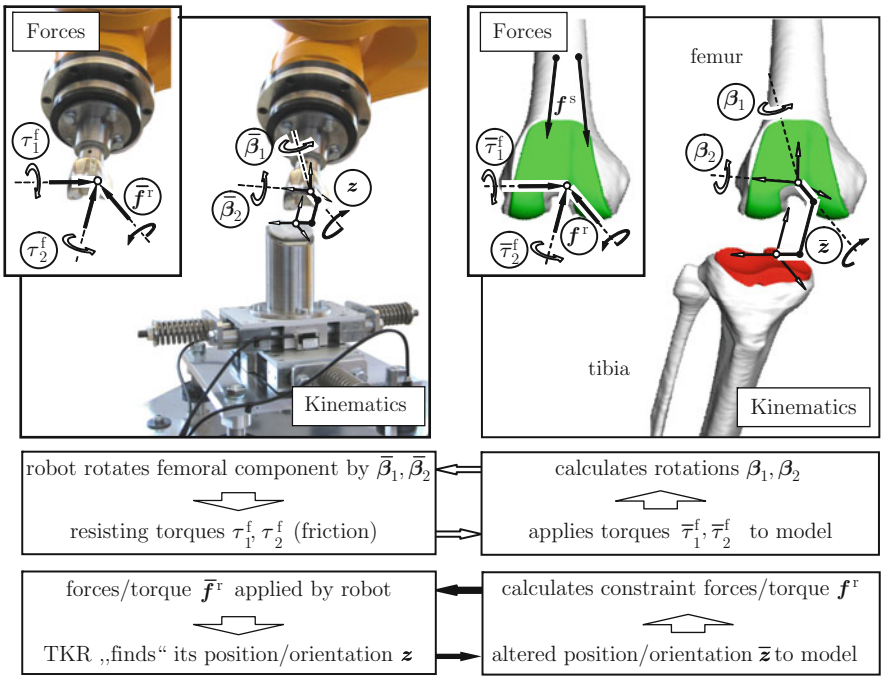



**Fig. 9.4** HiL simulation for testing TKRs with illustration of the exchanged values between the musculoskeletal multibody model (*right*) and the actuator system (*left*) at the kinematic and force levels

directions are the rotations of the femur with angles $\beta_1$ (internal/external rotation) and angle $\beta_2$ (flexion/extension) summarised in vector $\boldsymbol{\beta} \in \mathbb{R}^2$. The three translational directions and the remaining rotational direction of the femoral joint center with coordinates summarised in vector $\boldsymbol{z} \in \mathbb{R}^4$ are treated as the constrained directions.

### *9.3.3 Musculoskeletal Models*

In the HiL joint simulation the musculoskeletal model simulates the anatomical environment of the artificial joint which is not physically available. The objective of the model is to calculate the reaction forces $\boldsymbol{f}^r$ in the artificial joint for a given human motion. It also has to account for the soft tissue response during an instability event. Hence, capsular, ligament and muscle structures and their respective forces have to be incorporated into the model as well as geometric proportions and inertial properties of the skeletal system.

A musculoskeletal model consists of several skeletal segments linked to a kinematic chain. Depending on the joint under investigation and on the load case specific musculoskeletal models are built up. The models are implemented using the multibody simulation package SIMPACK (v8.9, Simpack AG, Gilching, Germany).

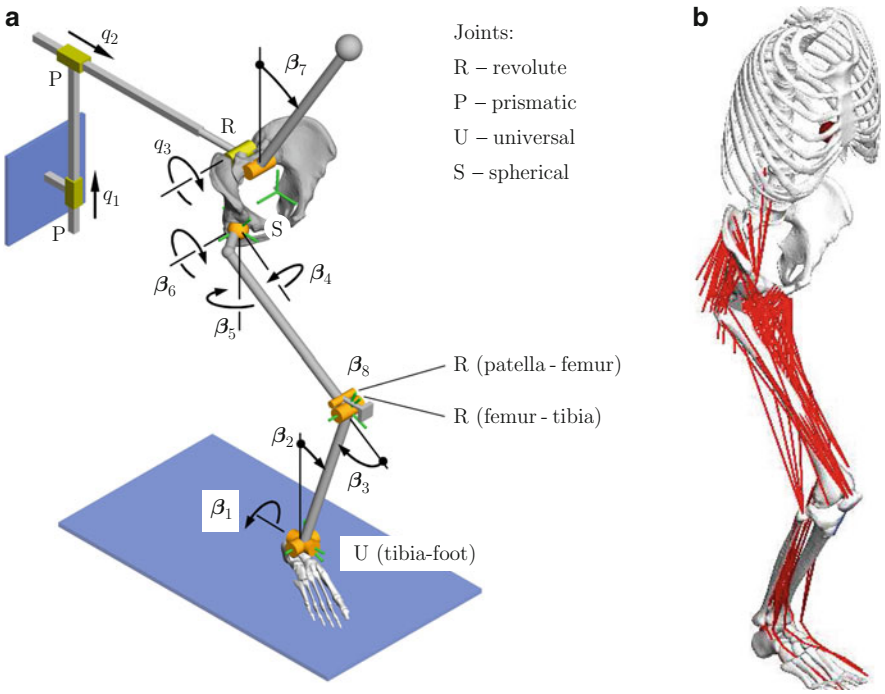



**Fig. 9.5** Multibody model of the lower extremity for THR testing. (**a**) Multibody topology. (**b**) Musculoskeletal model

## 9.4 HiL Testing a Total Hip Replacement

A human motion that is prone to dislocation of total hip replacements (THR) is a deep squat leading to a high flexion angle. For HiL simulation of this scenario, a musculoskeletal model is used that is symmetrical with respect to the sagittal plane [6].

### *9.4.1 Musculoskeletal Model for THR Testing*

According to Fig. 9.5, the musculoskeletal model used for these investigations represents the right lower extremity of the human body. It consists of the bone structures and the soft tissue comprising capsule, ligaments, and muscles.

#### 9.4.1.1 Coordinates

The kinematic chain consists of the foot assumed to be ground-fixed, tibia and fibula modeled as one rigid body, femur, and pelvis. The chain contains a universal

ankle joint (U), a revolute knee joint (R), and a hip joint. The hip joint is modelled by a kinematic subchain consisting of three orthogonal prismatic joints and three revolute joints with co-intersecting axes being equivalent to a spherical joint (S). According to Fig. 9.5, the revolute joints with rotation angles $\beta_4$, $\beta_5$, $\beta_6$ enable the free hip joint motions: internal/external rotation, flexion/extension, and abduction/adduction. The coordinates of the prismatic joints are constrained by the measurements $\bar{\boldsymbol{z}}$ provided by the robot, see Fig. 9.3.

The symmetry condition is achieved by means of a fictive planar joint in the sagittal plane, connecting the pelvis by two prismatic (P) and a revolute (R) joint with the ground. The constraint forces and torques of the planar joint represent the reaction forces between the modeled right lower extremity and its left counterpart that is not included in the model.

To the pelvis a single rigid body is attached by a revolute (R) joint representing the trunk. The patella is relevant for the direction of certain muscle forces influencing the hip motion. In the model the patella is connected with the femur by a revolute (R) joint. Thus, the multibody model has altogether eleven joint degrees of freedom and one kinematical loop. According to the Chebychev-Grübler-Kutzbach criterion, the overall degree of freedom of the model is $f = 5$.

If the kinematical loop is cut at the planar joint, the position of the spanning tree is described by the eight joint coordinates $\boldsymbol{\beta} = [\beta_1 \ \ldots \ \beta_8]^{\mathrm{T}}$. The cut planar joint leads to three implicit loop closure constraints for $\boldsymbol{\beta}$ at the position, velocity, and acceleration levels, respectively,

$$\boldsymbol{g}(\boldsymbol{\beta}, \bar{\boldsymbol{z}}) = \boldsymbol{0}, \tag{9.1}$$

$$\boldsymbol{G}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})\,\dot{\boldsymbol{\beta}} = \boldsymbol{0} \qquad \text{with} \qquad \boldsymbol{G} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\beta}} \in \mathbb{R}^{3,8}, \tag{9.2}$$

$$\boldsymbol{G}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})\,\ddot{\boldsymbol{\beta}} + \bar{\bar{\boldsymbol{\gamma}}}(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}, \bar{\boldsymbol{z}}) = \boldsymbol{0} \qquad \text{with} \qquad \bar{\bar{\boldsymbol{\gamma}}} = \dot{\boldsymbol{G}}\,\dot{\boldsymbol{\beta}} \in \mathbb{R}^3. \tag{9.3}$$

The constraints depend on the measured displacements $\bar{\boldsymbol{z}}$ in the constrained directions of the hip joint, see Fig. 9.3. Their time derivatives $\dot{\bar{\boldsymbol{z}}}$ and $\ddot{\bar{\boldsymbol{z}}}$ are neglected as the dynamics of the displacements in the constrained directions is not physically based but governed by the force controller of the robot. This assumption is considered to be acceptable as long as the displacements $\bar{\boldsymbol{z}}$ are small.

The motion of the system is described by $f = 5$ independent or minimal coordinates $\boldsymbol{q}$. Here, the three coordinates $q_1$, $q_2$, $q_3$ of the planar joint, the rotation angle of the trunk $\beta_7$, and the rotation angle of the patella $\beta_8$ are chosen, thus $\boldsymbol{q} = [q_1 \ \ q_2 \ \ q_3 \ \ \beta_7 \ \ \beta_8]^{\mathrm{T}}$. The joint coordinates $\boldsymbol{\beta}$ of the spanning tree can then be expressed in terms of the minimal coordinates $\boldsymbol{q}$ by means of the explicit loop closure conditions

$$\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{q}, \bar{\boldsymbol{z}}), \tag{9.4}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{J}(\boldsymbol{q}, \bar{\boldsymbol{z}})\, \dot{\boldsymbol{q}} \qquad \text{with} \qquad \boldsymbol{J} = \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{q}} \in \mathbb{R}^{8,5}, \tag{9.5}$$

$$\ddot{\boldsymbol{\beta}} = \boldsymbol{J}(\boldsymbol{q})\, \ddot{\boldsymbol{q}} + \bar{\bar{\boldsymbol{\eta}}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \bar{\boldsymbol{z}}) \quad \text{with} \quad \bar{\bar{\boldsymbol{\eta}}} = \dot{\boldsymbol{J}}\, \dot{\boldsymbol{q}} \in \mathbb{R}^{8}. \tag{9.6}$$

The free and constrained directions in the space of the joint coordinates $\boldsymbol{\beta}$ are orthogonal,

$$\boldsymbol{G}\boldsymbol{J} = \boldsymbol{0}. \tag{9.7}$$

#### 9.4.1.2 Soft Tissue Kinematics and Forces

In the multibody model, the soft tissue structures are included as force elements. Ligaments and joint capsules are described by passive, viscoelastic forces, while muscles are modeled by active force laws. The length and rate of length of the soft tissue elements appearing in the force laws are kinematically expressed in terms of the joint coordinates $\boldsymbol{\beta}$ and their time derivatives.

If a soft tissue structure $i$ acts along a straight line between two skeletal attachment points $P_i$ and $Q_i$ according to Fig. 9.6, its length $s_i$ and the length rate $\dot{s}_i$ are expressed by

$$s_i = \sqrt{\boldsymbol{s}_i^{\mathrm{T}}\, \boldsymbol{s}_i}\,, \qquad \dot{s}_i = \frac{\boldsymbol{s}_i^{\mathrm{T}}\, \dot{\boldsymbol{s}}_i}{s_i} \qquad \text{with} \qquad \boldsymbol{s}_i = \boldsymbol{r}_{Qi} - \boldsymbol{r}_{Pi}. \tag{9.8}$$

The position and velocity of points $P_i$ and $Q_i$ can be expressed in terms of the joint coordinates $\boldsymbol{\beta}$ of the spanning tree and the measured relative displacements of the implant components $\bar{\boldsymbol{z}}$, thus $\boldsymbol{r}_{Pi}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})$, $\boldsymbol{r}_{Qi}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})$, and

$$\dot{\boldsymbol{r}}_{Pi} = \boldsymbol{J}_{Pi}\, \dot{\boldsymbol{\beta}}, \quad \dot{\boldsymbol{r}}_{Qi} = \boldsymbol{J}_{Qi}\, \dot{\boldsymbol{\beta}} \quad \text{with} \quad \boldsymbol{J}_{Pi} = \frac{\partial \boldsymbol{r}_{Pi}}{\partial \boldsymbol{\beta}}, \quad \boldsymbol{J}_{Qi} = \frac{\partial \boldsymbol{r}_{Qi}}{\partial \boldsymbol{\beta}}. \tag{9.9}$$

The soft tissue length and its time derivative are then obtained in terms of $\boldsymbol{\beta}$ and $\bar{\boldsymbol{z}}$, thus $s_i(\boldsymbol{\beta}, \bar{\boldsymbol{z}})$ and

$$\dot{s}_i = \boldsymbol{J}_i^{\mathrm{s}}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})\, \dot{\boldsymbol{\beta}} \quad \text{with} \quad \boldsymbol{J}_i^{\mathrm{s}} = \frac{\boldsymbol{s}_i^{\mathrm{T}} \left(\boldsymbol{J}_{Qi} - \boldsymbol{J}_{Pi}\right)}{s_i} \equiv \frac{\partial s_i}{\partial \boldsymbol{\beta}} \in \mathbb{R}^{1,8}. \tag{9.10}$$

According to the force law to be applied, the soft tissue force depends on $s_i$, $\dot{s}_i$ and an activation input $u_i$, thus $f_i^{\mathrm{s}}(s_i, \dot{s}_i, u_i)$. Appropriate force laws are described in [1, 10]. By means of the Jacobi matrix $\boldsymbol{J}_i^{\mathrm{s}}$ from (9.10) the equivalent joint torques $\boldsymbol{\tau}_i^{\mathrm{s}} = [\tau_{i,1}^{\mathrm{s}} \ldots \tau_{i,8}^{\mathrm{s}}]^{\mathrm{T}}$ around the eight joint axes are obtained,

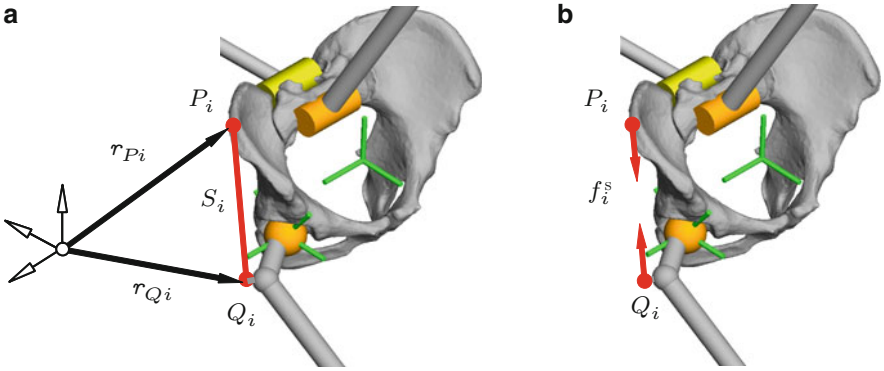



**Fig. 9.6** Soft tissue between two skeletal points $P_i$ and $Q_i$. (**a**) Geometry. (**b**) Soft tissue force

$$\boldsymbol{\tau}_i^{\mathrm{s}} = \boldsymbol{J}_i^{\mathrm{sT}} f_i^{\mathrm{s}}. \tag{9.11}$$

The kinematics of soft tissues that are extended over more than one joint is described in [1, 11].

#### 9.4.1.3 Equations of motion

The equations of motion of the multibody model shown in Fig. 9.5 are in terms of the joint coordinates $\boldsymbol{\beta} \in \mathbb{R}^8$

$$\boldsymbol{M}(\boldsymbol{\beta}, \bar{\boldsymbol{z}})\, \ddot{\boldsymbol{\beta}} = \boldsymbol{\tau}^{\mathrm{c}}\left(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}, \bar{\boldsymbol{z}}\right) + \boldsymbol{\tau}^{\mathrm{s}}\left(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}, \bar{\boldsymbol{z}}, \boldsymbol{u}\right) + \bar{\boldsymbol{\tau}}^{\mathrm{f}} + \boldsymbol{\tau}^{\mathrm{r}}. \tag{9.12}$$

The mass matrix $\boldsymbol{M} \in \mathbb{R}^{8,8}$ is obtained under the assumption that the soft tissue masses are added to the masses of the corresponding skeletal bodies. Dynamic wobbling of the muscle masses is neglected. The vector $\boldsymbol{\tau}^{\mathrm{c}} \in \mathbb{R}^8$ contains the torques of the centrifugal and Coriolis forces as well as gravity forces with respect to the joint axes. The vector $\boldsymbol{\tau}^{\mathrm{s}} \in \mathbb{R}^8$ is the sum of all soft tissue torques $\boldsymbol{\tau}_i^{\mathrm{s}}$ from (9.11) where $\boldsymbol{u}$ is the vector of all activation inputs. The vector $\bar{\boldsymbol{\tau}}^{\mathrm{f}}$ incorporates the resisting torques fed back from the robot into the model, see Fig. 9.3. The vector $\boldsymbol{\tau}^{\mathrm{r}} \in \mathbb{R}^8$ includes the torques of the constraint forces at the cut planar joint that are expressed by means of the Jacobi matrix $\boldsymbol{G} \in \mathbb{R}^{3,8}$ from (9.2) and the vector $\boldsymbol{\lambda} \in \mathbb{R}^3$ with the reaction force coordinates (Lagrange multipliers) of the planar joint,

$$\boldsymbol{\tau}^{\mathrm{r}} = \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{\beta})\, \boldsymbol{\lambda}. \tag{9.13}$$

The equations of motion in the joint coordinates $\boldsymbol{\beta}$ consist of the differential equations (9.12) and the loop closure constraints (9.1), (9.2), and (9.3).

#### 9.4.1.4 Inverse Dynamics

Dislocation scenarios are analysed for given human motions that are measured by a motion capture system. It is assumed that the independent coordinates $\boldsymbol{q}(t)$ are measured. The joint coordinates $\boldsymbol{\beta}(t)$ of the multibody model are then obtained from the explicit loop closure constraints (9.4), (9.5), and (9.6). In order to obtain the reaction forces in the hip joint appearing in Fig. 9.3, it is first necessary to get the active muscle forces associated with the given motion by means of an inverse dynamic calculation. Because of redundancy of the active muscle groups, the muscle forces cannot be uniquely determined. Another source of redundancy of the actuation forces is the closed kinematical loop of the model that represents human motions with both feet attached to the ground.

For inverse dynamics calculation, the soft tissue torques $\boldsymbol{\tau}^{\mathrm{s}}$ in (9.12) are split up into passive torques $\boldsymbol{\tau}^{\mathrm{p}}(\boldsymbol{\beta}, \dot{\boldsymbol{\beta}}, \bar{\boldsymbol{z}})$ due to viscoelastic forces and active torques $\boldsymbol{\tau}^{\mathrm{a}}$ due to active forces $\boldsymbol{f}^{\mathrm{a}}$, thus $\boldsymbol{\tau}^{\mathrm{s}} = \boldsymbol{\tau}^{\mathrm{p}} + \boldsymbol{\tau}^{\mathrm{a}}$. If there are $m$ active forces $\boldsymbol{f}_i^{\mathrm{a}}$, the overall active torques $\boldsymbol{\tau}^{\mathrm{a}}$ are

$$\boldsymbol{\tau}^{\mathrm{a}} = \boldsymbol{J}^{\mathrm{a}\,\mathrm{T}} \boldsymbol{f}^{\mathrm{a}} \quad \text{with} \quad \boldsymbol{J}^{\mathrm{a}} = \begin{bmatrix} \boldsymbol{J}_1^{\mathrm{a}} \\ \vdots \\ \boldsymbol{J}_m^{\mathrm{a}} \end{bmatrix} \in \mathbb{R}^{m,8}, \quad \boldsymbol{f}^{\mathrm{a}} = \begin{bmatrix} f_1^{\mathrm{a}} \\ \vdots \\ f_m^{\mathrm{a}} \end{bmatrix} \in \mathbb{R}^{m} \tag{9.14}$$

with the Jacobi matrices $\boldsymbol{J}_i^{\mathrm{a}}$ determined analogous to $\boldsymbol{J}_i^{\mathrm{s}}$ in (9.11).

For known joint coordinates $\boldsymbol{\beta}$ and their time derivatives, the active torques $\boldsymbol{\tau}^{\mathrm{a}}$ are obtained from (9.12). First the reaction force coordinates $\boldsymbol{\lambda}$ are eliminated by left-multiplicating (9.12) with the transposed Jacobi matrix $\boldsymbol{J}$ of the explicit loop closure constraints (9.5) under consideration of the orthogonality (9.7). Together with (9.14), this leads to an underdetermined system of $f = 5$ linear equations for the $m > f$ active muscle forces $\boldsymbol{f}^{\mathrm{a}}$,

$$\boldsymbol{A}\,\boldsymbol{f}^{\mathrm{a}} = \boldsymbol{b} \quad \text{with} \quad \boldsymbol{A} = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{J}^{\mathrm{aT}}, \quad \boldsymbol{b} = \boldsymbol{J}^{\mathrm{T}} \left( \boldsymbol{M}\,\ddot{\boldsymbol{\beta}} - \boldsymbol{\tau}^{\mathrm{c}} - \boldsymbol{\tau}^{\mathrm{p}} - \overline{\boldsymbol{\tau}}^{\mathrm{f}} \right). \tag{9.15}$$

A common way to find a physiologically based solution for the active forces $\boldsymbol{f}^{\mathrm{a}}$ is to regard the linear system of equations (9.15) as an equality constraint of a static optimisation problem minimising a cost function $I(\boldsymbol{f}^{\mathrm{a}})$. A typical definition is a quadratic cost function

$$I(\boldsymbol{f}^{\mathrm{a}}) \equiv \boldsymbol{f}^{\mathrm{a}\,\mathrm{T}} \boldsymbol{Q}\, \boldsymbol{f}^{\mathrm{a}} \stackrel{!}{=} \min_{\boldsymbol{f}^{\mathrm{a}}} \tag{9.16}$$

with a positive definite weighting matrix $\boldsymbol{Q} \in \mathbb{R}^{m,m}$. Examples for the definition of $\boldsymbol{Q}$ and also for other cost functions are discussed in [1]. With the obtained active forces $\boldsymbol{f}^{\mathrm{a}}$ the reaction force components $\boldsymbol{f}^{\mathrm{r}}$ in the actual position of the hip joint are calculated.

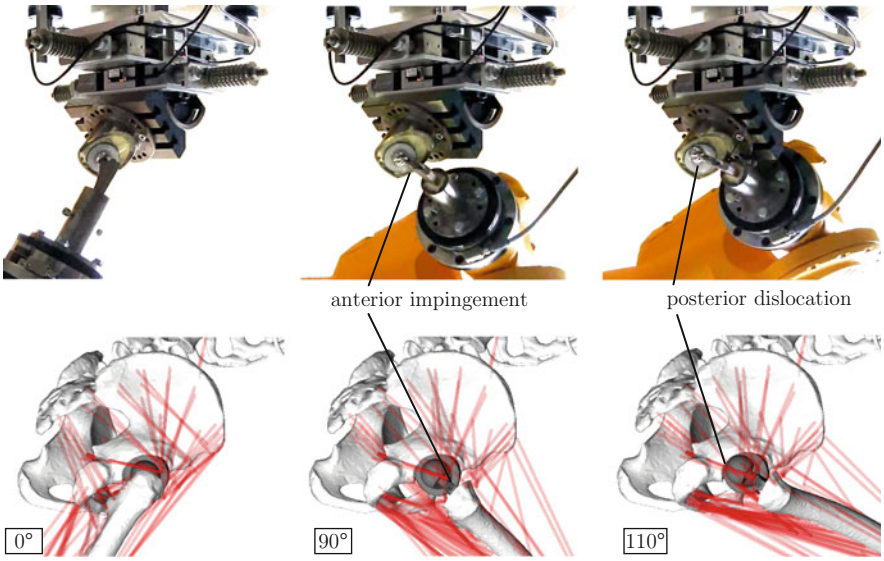



**Fig. 9.7** Motion of the total hip replacement in comparison with the multibody model during HiL simulation of a deep squat with flexion angles changing from 0° (standing position) over 90° to 110° (45° inclination of the acetabular cup)

### *9.4.2 Results of THR Testing*

As an example a deep squat leading to a high flexion angle up to about 110° and an adduction angle of about 15° is considered. The outcome of the HiL simulation with the motion of the artificial hip joint (Alloclassic System, Zimmer GmbH, Winterthur, Switzerland) in comparison to the multibody model is shown in Fig. 9.7. Shortly after the 90° position, an anterior impingement (physical contact between prosthetic neck and anterior rim of the acetabular cup) occurs, followed by a posterior dislocation of the femoral head in the 110° position. The comparison between two orientations of the acetabular cup (45° and 60° inclination together with 0° anteversion) showed an anterior impingement at about 90° flexion in both cases while a dislocation of the femoral head occurred at higher flexion angles.

## 9.5 HiL Testing of a Total Knee Replacement

HiL testing of a total knee replacement is shown for a passive flexion movement whereby different conditions of the anterior cruciate ligament (ACL) are considered with respect to the stability of the TKR.

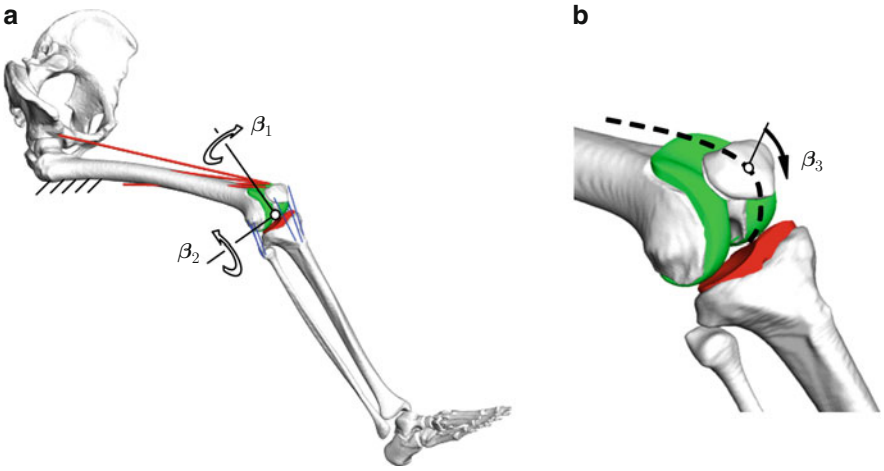



**Fig. 9.8** Musculoskeletal model of the lower extremity for TKR testing. (**a**) Multibody topology with tibio-femoral joint. (**b**) Patello-femoral joint with one degree of freedom

### *9.5.1 Musculoskeletal Model for TKR Testing*

Adapted to the load case considered in the following, the musculoskeletal model represents the lower right extremity of the human body with the TKR implanted (Fig. 9.8).

#### 9.5.1.1 Coordinates

The model has an open-chain topology with pelvis and femur kept fixed, the tibia connected with the femur by the tibio-femoral joint and the patella connected with the femur by the patello-femoral joint. The ankle joint is kept fixed. The tibio-femoral joint is modelled by a kinematic subchain consisting of three orthogonal prismatic joints and three revolute joints with co-intersecting axes. According to Fig. 9.4, two revolute joints correspond to the free internal/external rotation and flexion motion described by two rotation angles $\beta_1$ and $\beta_2$ while the coordinates of the four other joints are constrained by the measurements $\bar{\boldsymbol{z}}$ provided by the robot. The patello-femoral joint is modelled with one translational degree of freedom. The patella is bound on a spatial trajectory with respect to the femur whereby its position is described by the arc length $\beta_3$.

#### 9.5.1.2 Soft Tissue Kinematics and Forces

Soft tissue and muscle forces are calculated in analogy to the musculoskeletal model for THR simulation in Sect. 9.4.1. Ligament structures of both joints are represented as nonlinear spring-damper combinations following force-displacement characteristics

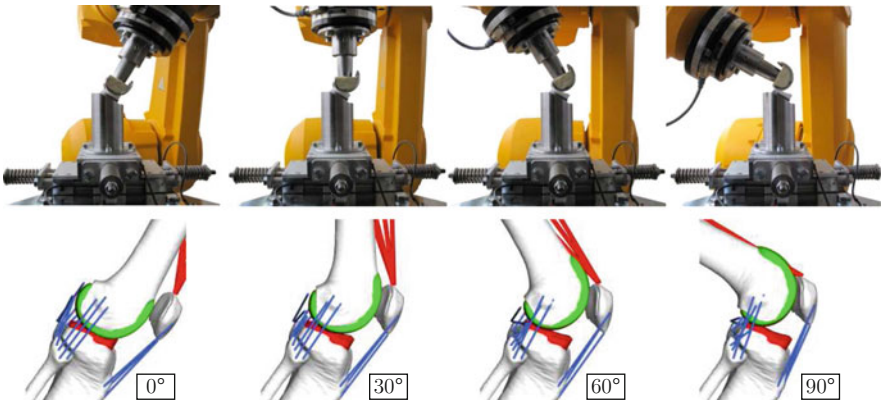



**Fig. 9.9** Motion of the total hip replacement in comparison with the multibody model during HiL simulation of a bicondylar unconstrained TKR without ACL bundles showing conformity between positioning of robot and multibody model at different flexion angles

derived from in vitro measurements [8]. Moreover, the four quadriceps muscle components (Mm. rectus femoris, vastus medialis, lateralis and intermedius muscles) are included by Hill type muscle elements [10] whereas only passive muscle forces are considered within this example.

### *9.5.2 Results of TKR Testing*

A passive flexion movement of a bicondylar unconstrained TKR (Multigen Plus knee, Lima Lto, San Daniele, Italy) is simulated with two different conditions of the anterior cruciate ligament (ACL) implemented into the multibody model: with and without all ACL bundles [5]. In most cases all ACL bundles are resected during implantation of bicondylar TKRs due to implant design. The considered load case emulates a postoperative passive mobilisation of the knee joint by a physical therapist without muscle activity.

The HiL simulation is carried out for both conditions of the anterior cruciate ligament (ACL) from 0° to 90° flexion at constant internal/external rotation of 0°. According to the given angles and the reaction forces/torques calculated by the multibody model, the robot rotates the femoral component and applies the load to the tibial component. By feeding back the measurements $\bar{\boldsymbol{z}}$ to the multibody model, the tibia was shifted and orientated in the constrained directions with respect to the femur influencing the elongation of the incorporated ligaments (Fig. 9.9). Due to the interdependency between measurements and calculated reaction forces by the elongation of ligaments, the different conditions of the ACL (with and without ACL bundles) result into different outcomes of the relative displacement of the femoral component with respect to the tibial polyethylene liner. Considering

anterior/posterior translation, the femoral component tends to shift in anterior direction at flexion angles below 20° when the ACL bundles are incorporated. Contrarily, the femoral component shifts only in posterior direction when all ACL bundles are removed.

## 9.6 Conclusion

The presented approach of HiL simulation provides a highly flexible, state-of-the-art test system. It allows analysis and comparison of different joint replacement systems with respect to their behaviour in case of joint instability under physiological and reproducible load conditions. This includes the complex soft tissue response during such an event. Hence, it is possible to gain visual insights into the process of instability events due to the attachment of the real implant components on the physical setup and the post-processing of the multibody model. This is exemplarily demonstrated for a THR by simulating a deep squat movement and for a TKR by simulating a flexion movement with different soft tissue conditions. The HiL simulations show that the change in ligament structures alters the load situation and, thus, the kinematics of the artificial joint.

To further develop the HiL test method more instability-associated manoeuvres for both THRs and TKRs will be investigated under different ligament and muscle conditions. Subsequently, influencing factors such as implant design and positioning as well as soft tissue resection can be investigated systematically on the basis of realistic load case scenarios. Due to the incorporation of real contact conditions within this approach, the outcomes of the measured relative motion provide a basis for evaluating the performance of each THR or TKR system, respectively. The results yield more precise insights and data of the process leading to artificial joint instability. In this manner, the proposed HiL simulations can assist in the improvement of endoprosthetic design, preoperative planning, the choice of appropriate endoprosthetic components, and surgical treatment for given bony and soft tissue structures in case of primary, revision, and tumour surgery.

## References

1. Ambrósio JAC, da Silva MP (2005) A biomechanical multibody model with a detailed locomotion muscle apparatus. In: Ambrósio JAC (ed) Advances in computational multibody systems. Springer, Dordrecht, pp 155–184
2. Bozic KJ, Kurtz SM, Lau E, Ong K, Chiu V, Vail TP, Rubash H, Berry DJ (2010) The epidemiology of revision total knee arthroplasty in the United States. Clin Orthop Relat Res 468(1):45–51
3. Bozic KJ, Kurtz SM, Lau E, Ong K, Vail TP, Berry DJ (2009) The epidemiology of revision total hip arthroplasty in the United States. J Bone Joint Surg Am vol 91(1):128–133

4. Herrmann S, Kaehler M, Souffrant R, Rachholz R, Zierath J, Kluess D, Mittelmeier W, Woernle C, Bader R (2011) HiL simulation in biomechanics: a new approach for testing total joint replacements. Comput Methods Programs Biomed 105(2):109–119
5. Herrmann S, Woernle C, Kähler M, Rachholz R, Souffrant R, Zierath J, Kluess D, Bader R (2012) HiL simulation for testing joint stability after total knee arthroplasty. Multibody Sys Dyn. 28:55–67
6. Kähler M, Rachholz R, Herrmann S, Zierath J, Souffrant R, Kluess D, Bader R, Woernle C (2010) Development of a biomechanical multibody model for the hardware-in-the-loop simulation of total hip endoprostheses. In: Mikkola A (ed) Proceedings of the 1st Joint international conference on multibody system dynamics. Lappeenranta (Finland)
7. Kähler M, Woernle C, Bader R (2009) Hardware-in-the-loop simulaton of constraint elements in mechanical systems. In: Kecskeméthy A, Müller A (eds) Computational kinematics. Springer, Berlin, pp 159–166
8. Lehner S (2008) Entwicklung und Validierung biomechanischer Computermodelle und deren Einsatz in der Sportwissenschaft. Ph.D. thesis, Universität Koblenz-Landau
9. Siciliano B, Sciavicco L, Villani L, Oriolo G (2008) Robotics: modelling, planning and control. Springer, Berlin
10. Spägele T, Kistner A, Gollhofer A (1999) Modelling, simulation and optimisation of a human vertical jump. J Biomech 32:521–530
11. Woernle C, Kähler M, Rachholz R, Herrmann S, Zierath J, Souffrant R, Bader R (2010) Robot-based HiL test of joint endoprostheses. In: Lenarčič J, Stanišić MM (eds) Advances in robot kinematics. Springer, Berlin, pp 521–528

# Chapter 10
# Dynamics and Control of the Biped Robot Lola

**Thomas Buschmann, Valerio Favot, Markus Schwienbacher, Alexander Ewald, and Heinz Ulbrich**

**Abstract** This paper gives an overview of the dynamics and control of the humanoid walking robot Lola. A brief analysis of the robot's multibody dynamics motivates our approach to biped walking control. After a brief description of the robot Lola we outline the architecture of its hierarchical walking control system. We also present the real-time planning method for center of gravity trajectories and the model-based hybrid position/force control module that acts as a basis for the stabilizing walking controller.

## 10.1 Introduction

From an economic point of view, the development of humanoid robots is motivated by the broad potential for service applications. A human-like shape and (some) human-like skills promise universal service robots in the original sense of Karel Čapeks play "Rossum's Universal Robots". For these machines, biped locomotion is an essential skill. A further reason for developing humanoid robots is the fact that legged machines potentially are more flexible than wheeled or tracked vehicles, since they can step over or onto obstacles instead of driving around them. During the past decades, significant research efforts have been aimed at developing human-like legged locomotion.

The fastest biped robots are developed by companies, such as Honda's Asimo [1], Toyota's running robot [2] and the robot Petman from Boston Dynamics [3]. The robots of the HRP-series are developed by Japan's AIST in cooperation with Kawada industries (HRP-2 [4], HRP-3 [5] and HRP-4 [6]). Other notable developments are the Wabian-2 [7] and HUBO [8] robots.

T. Buschmann (✉) • V. Favot • M. Schwienbacher • A. Ewald • H. Ulbrich
Institute of Applied Mechanics, Technische Universität München, Bolzmannstr. 15, Garching 85747, Germany
e-mail: thomas.buschmann@tum.de; favot@amm.mw.tum.de; schwienbacher@amm.mw.tum.de; ewald@amm.mw.tum.de; ulbrich@amm.mw.tum.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_10, 

Currently, the fastest bipeds are developed by Honda. By improving both the hardware design and walking pattern generation and control Takenaka et al. achieved very fast running with a Honda-prototype at 10 km/h [9–12]. This technology was integrated into the newest Asimo-robot which is capable of running at 9 km/h [13]. Boston Dynamics has shown very fast walking with the DARPA-funded Petman prototype [3], but the control algorithms and details of the hardware design are regarded as company secrets.

In the following we give an overview of the walking control system developed for our biped robot Lola. This article contains material from previous publications [14–16], with an additional section on walking dynamics.

## 10.2 System Overview

Lola is 180 cm tall and weighs approximately 60 kg. Figure 10.1 shows a picture of Lola, the kinematic structure and a table of the actuated joints. The kinematic structure is based on anthropometric data and features active toe and pelvis joints. Lola was designed for light weight, high effective stiffness and low leg inertia. While major segments are very stiff, Lola's feet are equipped with viscoelastic contact elements that are deformed during walking and have a strong influence on walking dynamics. The actuators are based on brushless kit motors that are combined with reduction gears and sensors to compact drive units with a high torque density. The robot is equipped with incremental encoders on the motor side and absolute encoders on the link side. An inertial measurement unit (IMU) is mounted on the upper body and six axis force/torque sensors (FTS) are integrated into the feet. Figure 10.2 shows Lola's electronics architecture. Nine local controllers implement protocols and interfaces required for the various sensors and servo controllers and communicate with the central onboard PC via the fast real-time Ethernet communication system Sercos-III. A detailed description of the mechanical design, sensors and electronics is given in [15, 17].

## 10.3 Aspects of Biped Robot Dynamics

Humanoid robots such as Lola are complex mechatronic systems and there always is a trade-off between modeling depth and accuracy on the one hand and computational efficiency on the other hand. Experience with the robots Lola and Johnnie has shown that the major dynamical effects on biped locomotion for this type of robot are, in decreasing order of importance:


| Joint | DoF |
|---|---|
| Head | $1 \times 3$ |
| Shoulder | $2 \times 2$ |
| Elbow | $2 \times 1$ |
| Pelvis | $1 \times 2$ |
| Hip | $2 \times 3$ |
| Knee | $2 \times 1$ |
| Ankle | $2 \times 2$ |
| Toe | $2 \times 1$ |
| Total | 25 |

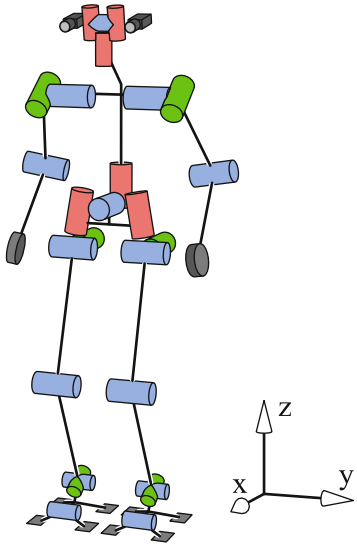



**Fig. 10.1** The biped robot Lola (*left*) has 25 actively driven joints. The kinematic structure is shown on the *right*

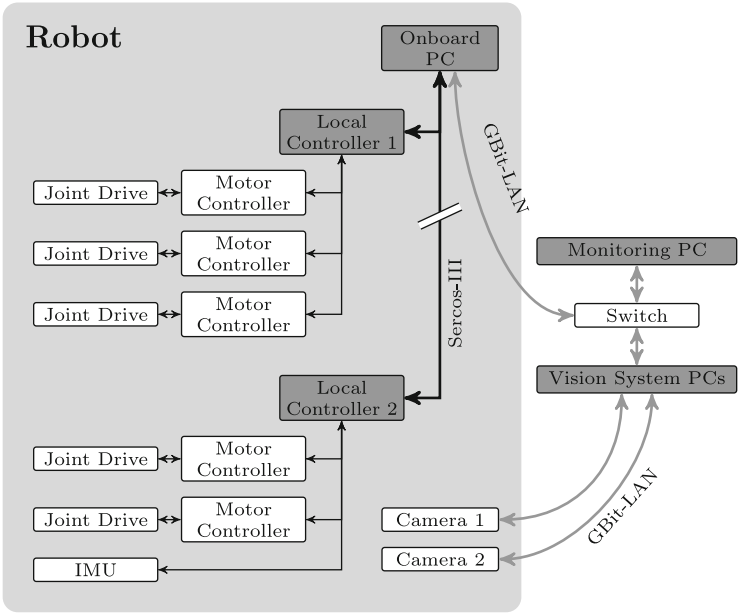



**Fig. 10.2** Lola has a sensor actuator network with nine local controllers. Sensors are read out by both motor controllers and local controllers. The onboard PC is connected to the local controllers via the fast real-time Ethernet-based Sercos-III system. Trajectory planning and stabilizing control run on the onboard PC

1. Rigid body mechanics of segments, motor shafts and gears,
2. Unilateral, compliant foot-ground contact,
3. Gear elasticity,
4. Drive friction,
5. Non-linear kinematics in ankle and knee joints and
6. Electrical motor dynamics.

Items (1), (3), (4) and (5) lead to a set of second order ODEs describing the multibody dynamics. We use the Newton-Euler method to obtain a minimal coordinate representation of the system.

The contact model (2) adds a set of first order ODEs describing the deformation of the contact layer and complementarity conditions for unilateral contacts and coulomb friction. Finally, the electrical motor dynamics (6) add another set of first order ODEs [18].

To motivate the design of our control system, we focus on the first two effects, leading to the following system equations:

$$\begin{pmatrix} \mathbf{M}_{TT} & \mathbf{M}_{TJ} \\ \mathbf{M}_{JT} & \mathbf{M}_{JJ} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}}_T \\ \ddot{\mathbf{q}}_J \end{pmatrix} + \begin{pmatrix} \mathbf{h}_T \\ \mathbf{h}_J \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{W}_{\boldsymbol{\tau},J} \end{pmatrix} \boldsymbol{\tau} + \begin{pmatrix} \mathbf{W}_{\boldsymbol{\lambda},T} \\ \mathbf{W}_{\boldsymbol{\lambda},J} \end{pmatrix} \boldsymbol{\lambda} \tag{10.1}$$

The minimal coordinates are denoted by $\mathbf{q}^T = (\mathbf{q}_T^T, \mathbf{q}_J^T)$, the lower index $T$ denotes torso DoFs and the lower index $J$ joint angles. The actuator torques are denoted by $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$ are the contact forces and all other forces are summed up in the vector $\mathbf{h}$. The matrices $\mathbf{W}_\tau$ and $\mathbf{W}_\lambda$ are Jacobians for projecting actuator and contact forces into the unconstrained directions.

The Jacobian for the actuator forces is rank deficient, i.e., the robot is underactuated. This means that stable walking using a simple joint-level trajectory tracking controller is very difficult and not robust, since it is not possible to directly control the upper body DoFs.

On the other hand, a high-gain joint control loop has the advantage of canceling gear friction, which is temperature, load and speed dependent and difficult to identify accurately. A hierarchical approach therefore seems promising. A low-level joint tracking controller can provide robustness against modeling errors and simplify robot dynamics. Higher layers can build on this to provide robustness against disturbances and deal with the problem of underactuation.

As a limiting case, we can assume perfect tracking for the joint position controller and replace the joint angles $\mathbf{q}_J$ in (10.1) with the desired joint angles $\mathbf{q}_{J,d}$. Since the joint angles are now known, we can omit their equations of motion (EoMs), leaving us with just six equations:

$$\mathbf{M}_{TT}\, \ddot{\mathbf{q}}_T + \mathbf{h}_J^\star = \mathbf{W}_{\lambda,T} \boldsymbol{\lambda} \tag{10.2}$$

The modified vector $\mathbf{h}_J^\star$ contains the forces due to the joint angle motion $\mathbf{q}_J$, which acts as a parametric excitation. Obviously, we can control the unactuated DoFs $\mathbf{q}_T$ and thereby stabilize the robot by manipulating the contact forces $\boldsymbol{\lambda}$. This observation motivates our hierarchical control system that stabilizes the overall

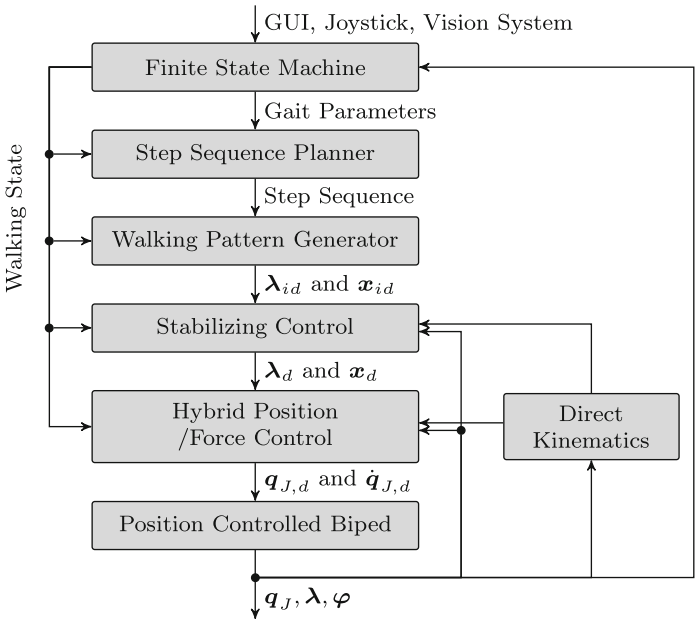



**Fig. 10.3** The architecture of Lola's walking control system. $\boldsymbol{\lambda}$ are contact forces, $\boldsymbol{x}$ task-space coordinates, $\mathbf{q}_J$ joint angles and $\boldsymbol{\varphi}$ is the upper body orientation. Indices *id* and *d* indicate ideal and modified references, no index is used for sensor data

system dynamics by controlling a subset of contact forces via an underlying position control loop, while at the same time tracking a subset of task-space trajectories.

## 10.4 Control System

This section describes Lola's control system architecture, the planning algorithm for center of gravity trajectories and the hybrid position/force control method.

### *10.4.1 Control System Architecture*

Lola's controller is designed as a hierarchical system whose individual components are selected and parameterized by a finite state machine. The structure of the system is shown in Fig. 10.3.

The model-based design makes the system applicable to any position-controlled biped with inertial and force sensing. Adaptation of the walking controller to a new robot mainly consists of providing a description of the robot as a multibody system. After that, only a small number of parameters must be adjusted in experiments to

account for different sensor characteristics, etc. We have successfully used the system in experiments with the robots Lola and Johnnie [19].

The desired walking behavior is input via a graphical user interface (GUI) or a joystick. We have also combined the walking controller with a computer vision system to demonstrate autonomous locomotion in unknown environments [20].

The user input, typically a desired walking speed or step length, is converted into a sequence of steps. The step sequence describes both the footstep locations and other critical gait parameters such as step duration and ground clearance for the swing leg. Continuous walking patterns and contact force trajectories are calculated online based on the step sequence (see Sect. 10.4.2). A stabilizing controller modifies the reference trajectories based on IMU and FTS feedback. It compensates modeling errors, disturbances and uneven terrain. The modified force and position trajectories are converted to desired joint trajectories by a hybrid position/force control. The lowest level in the control hierarchy is a high-gain joint position control loop.

### *10.4.2 Real-Time CoG-Trajectory Planning*

Lola's reference trajectories are planned in task-space. This section outlines the method for calculating center of gravity (CoG) trajectories, a central component of the online planning system.

There are three major difficulties in walking pattern generation for biped robots: First, the unilateral foot-ground contact leads to inequality constraints for the physically feasible contact forces. Second, the system is inherently unstable and small disturbances can lead to an exponential divergence of the trajectories of an unstabilized robot. Third, the complexity of a full multibody model means that a naive application of, e.g., an optimal control approach to planning is currently not suitable for real-time control.

To enable real-time planning we therefore use a reduced robot model consisting of three point masses (see Fig. 10.4). Two masses are used to approximate the inertial effects of the legs while the third mass accounts for the rest of the robot. This model gives significantly better results than the more widely used inverted pendulum model with one point mass.

The basic algorithm for calculating CoG-trajectories is a two-stage process. First, a physically feasible contact force trajectory that minimizes the rate of change is calculated by quadratic programming. Then, a the CoG-trajectory is obtained by solving a boundary value problem (BVP) for the reduced robot model [21].

The initial position and velocity for the BVP is given by the current references for the CoG position and velocity. The desired state at the end of the planning horizon is set to the initial state of a periodic reference gait. This periodic reference is itself calculated using the two-stage process described above, where the boundary

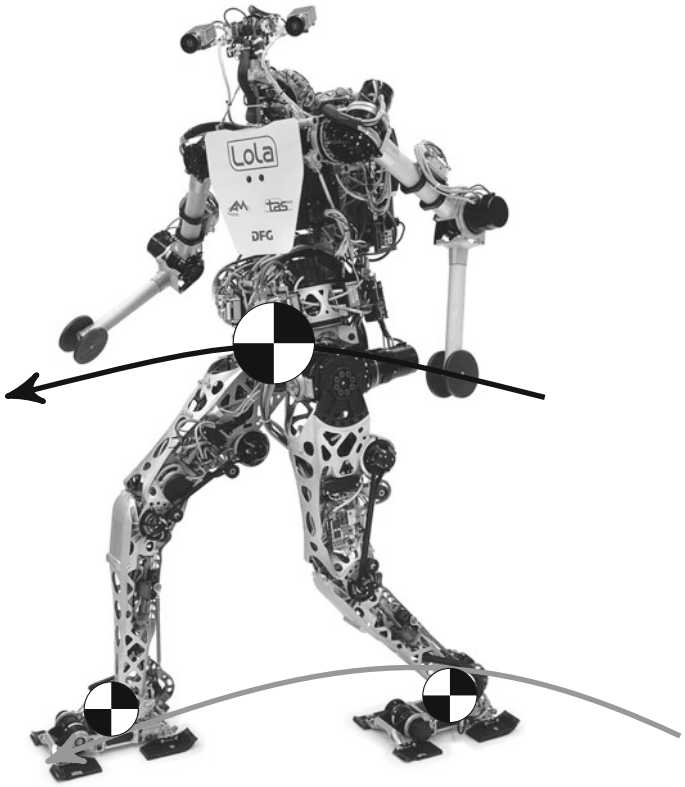



**Fig. 10.4** Visualization of the reduced robot model for real-time planning. Masses at the feet model the leg dynamics, a third point mass represents the remaining mass. The gray foot trajectory is assumed to be known, the black trajectory for the "upper body" mass is determined using the proposed method

conditions are set to ensure a periodic solution. The boundary conditions guarantee a stable trajectory by forcing convergence to a periodic and therefore stable orbit. Using periodic reference steps has previously been proposed by Tajima [2], Takenaka [10] and others.

#### 10.4.2.1 Solving the Boundary Value Problem

The EoM for the simplified robot model in the lateral plane are given by:

$$mz\ddot{y} - m(g + \ddot{z})y = -(T_x + \Delta T_{x,\text{leg}}) \tag{10.3}$$

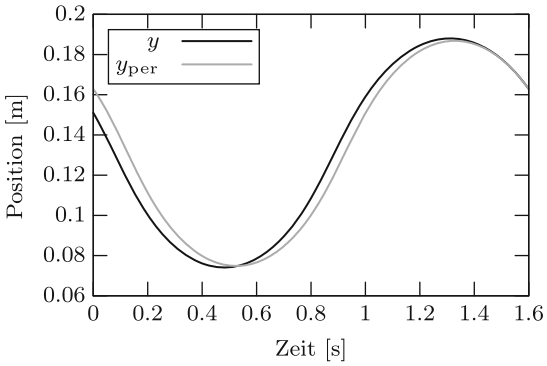



**Fig. 10.5** Example of a lateral-plane trajectory calculated with the proposed method. The gray line shows the periodic reference $y_{\text{per}}$ and the black line $y$ the reference for next two steps

Here $m$ denotes the "upper body" mass, $y$ is the lateral position of this point, $g$ is the gravity acceleration, $T_x$ the contact moment about the $x$-axis and $\Delta T_{x,\text{leg}}$ is the moment generated by the legs (see [21]).

This is a simple, linear EoM. However, since the coefficients are not constant and the right hand side can be quite complex, there is no closed form solution in the general case. We therefore propose solving the BVP using a collocation method [22] with cubic splines as basis functions. This approach is very flexible and simple to implement and gives sufficiently accurate results. The collocation method reduces (10.3) to a linear algebraic equation that can easily be solved in real-time. The entire planning process takes less than 1 ms on Lola's onboard computer.

Equation (10.3) is a second order ordinary differential equation. For a given right hand side, it is therefore in general not possible to satisfy boundary conditions for positions and velocities at the beginning and at the end of the trajectory. Two additional free parameters are needed to obtain a well-posed problem.

We propose using two parameters $\gamma_1, \gamma_2$ that act as weights for shape functions that modify the optimal contact force trajectory in order to satisfy the boundary conditions [21]. The set of equations for calculating the unknown spline parameters $\mathbf{p}$ for the approximate solution of the EoM and the additional parameters $\gamma_1, \gamma_2$ is given by:

$$\mathbf{B}\mathbf{p} + \mathbf{b}_0 = \mathbf{T}_x + \boldsymbol{\Delta}\mathbf{T}_x(\gamma_1, \gamma_2) \tag{10.4}$$

$$\mathbf{G}\mathbf{p} + \mathbf{g}_0 = \mathbf{0} \tag{10.5}$$

The first line is obtained by requiring that the approximate solution satisfy the EoM (10.3). The vector $\mathbf{T}_x$ contains the values of the ideal contact moments at the

collocation points. The modification of the contact moments $\Delta\mathbf{T}_x(\gamma_1, \gamma_2)$ is linear in the unknown parameters $\gamma_i$. The second line contains conditions for the position and velocity at the beginning and end of the trajectory.

Figure 10.5 shows a solution obtained with this method (black line) together with a periodic reference (gray line) toward which it converges.

Note that the mass and position of the "upper body" in the reduced model (10.3) are *not* equivalent to the actual mass and position of the robot's torso. Rather, the distribution is chosen to minimize the difference of the contact forces calculated using the full multibody model. The reference CoG for the walking controller is set to the CoG of the reduced order model.

### *10.4.3 Hybrid Force/Position Control*

The stabilizing control is based on manipulating a subset of contact forces while tracking reference trajectories described in task-space coordinates. To control the contact forces, the reference trajectories are modified along some task-space dimensions. That is, a hybrid position/force control with inner position control loop is used. A related control approach was previously proposed by [23]. See [24] for a more complete list of related work.

The subsets of task space trajectories ($\mathbf{x}$) and contact forces ($\boldsymbol{\lambda}$) that should be tracked are selected using binary matrices $\mathbf{S}_x, \mathbf{S}_\lambda$. The control law is derived from the chosen error dynamics:

$$\mathbf{S}_\lambda \left(\Delta\dot{\boldsymbol{\lambda}} + \mathbf{K}_\lambda \, \Delta\boldsymbol{\lambda}\right) = \mathbf{0} \tag{10.6}$$

The contact is modeled using a set of linearly elastic point contacts. This leads to a linear relationship between $\dot{\mathbf{x}}, \dot{\boldsymbol{\lambda}}$ and the generalized velocities $\dot{\mathbf{q}}$:

$$\dot{\boldsymbol{\lambda}} = \left(\nabla_q \boldsymbol{\lambda}\right)\dot{\mathbf{q}} \tag{10.7}$$

$$\dot{\mathbf{x}} = \left(\nabla_q \mathbf{x}\right)\dot{\mathbf{q}} \tag{10.8}$$

From these equation, we derive a modification of the ideal reference trajectories by $\Delta\,\dot{\mathbf{x}}_\lambda$ for controlling the contact forces without modifying the position controlled task space dimensions $\mathbf{S}_x\mathbf{x}$:

$$\begin{aligned}\Delta\,\dot{\mathbf{x}}_\lambda = \Big\{ [\mathbf{S}_\lambda \left(\nabla_q \boldsymbol{\lambda}\right)(\bar{\mathbf{S}}_x \left(\nabla_q \mathbf{x}\right))^{\#}]^{\#} \, \boldsymbol{\alpha}_\lambda \, \mathbf{S}_\lambda \left[\mathbf{K}_{\lambda,\mathrm{FF}} \, \dot{\boldsymbol{\lambda}}_{d1} + \mathbf{K}_\lambda \left(\boldsymbol{\lambda}_d - \boldsymbol{\lambda}\right)\right] \Big\} \\ + \boldsymbol{\alpha}_x \left[\bar{\mathbf{S}}_x \, \mathbf{K}_{\lambda x} \left(\mathbf{x}_d - \mathbf{x}\right)\right]\end{aligned} \tag{10.9}$$

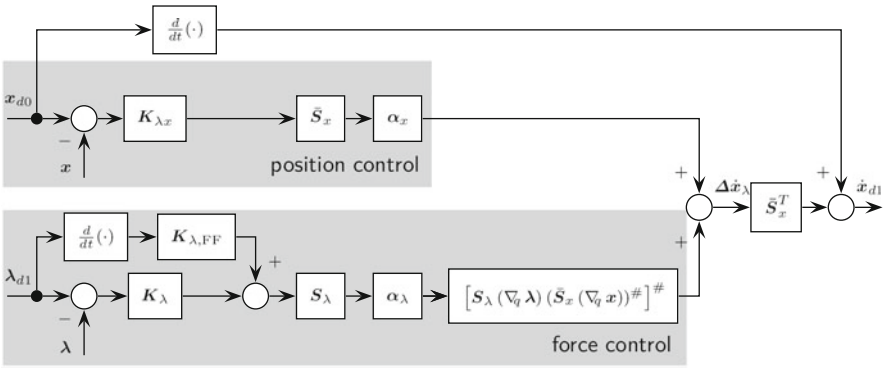



**Fig. 10.6** Proposed method of modifying task-space trajectories for hybrid position/force control

Here $\bar{\mathbf{S}}_i$ is the complement of $\mathbf{S}_i$ and $\mathbf{K}_{\lambda,\mathrm{FF}}$ is a gain factor. The matrices $\boldsymbol{\alpha}_\lambda, \boldsymbol{\alpha}_x$ are gains for blending between position and force control, which is used for disabling force control for the swing leg. Figure 10.6 illustrates the method for calculating the trajectory modification $\boldsymbol{\Delta}\dot{\mathbf{x}}_\lambda$ . See [24] for a more detailed derivation of the control law and a list of related work in walking and interaction control.

## 10.5 Experimental Results

This section summarizes some of the experimental results obtained with the robot Lola.

One of our research goals is to improve the walking speed of biped robots. The maximum walking speed we have demonstrated in experiments with Lola is approximately 3.6 km/h. Figure 10.7 shows Lola walking at 3.5 km/h. This is slow compared to human walking, but still makes Lola one of the fastest electrically driven biped walkers. However, both Honda and Toyota have shown significantly faster running and Boston Dynamics has shown faster walking with a hydraulic biped (see [16] for a more detailed discussion).

Autonomous, vision guided navigation has been a second area of research. Using a vision system developed at the Institute for Autonomous Systems Technology, University of the Bundeswehr Munich, we have demonstrated autonomous navigation in unknown environments using only onboard vision (see Fig. 10.8.and [20]).

More recently, we have been addressing the issue of locomotion over rough, unmodelled terrain. Using a reactive replanning strategy Lola is capable of walking over 4 cm high, unmodelled obstacles (see Fig. 10.9).

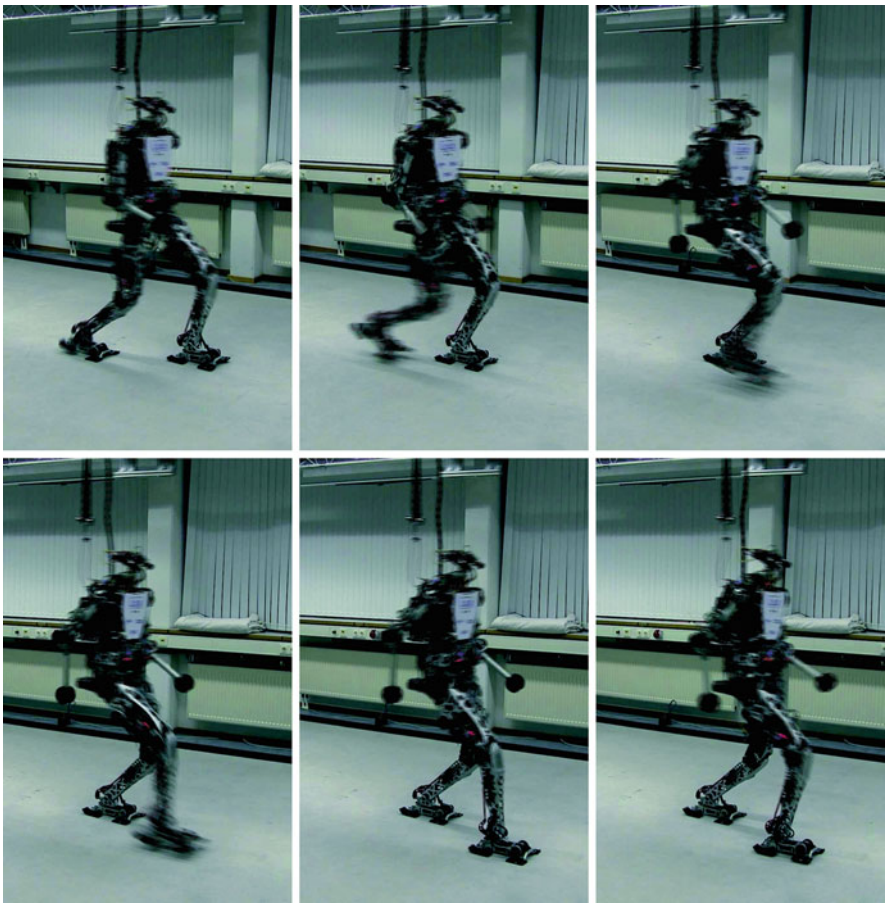


**Fig. 10.7** Lola walking at 3.5 km/h

**Fig. 10.8** Lola autonomously navigating an unknown environment [20]

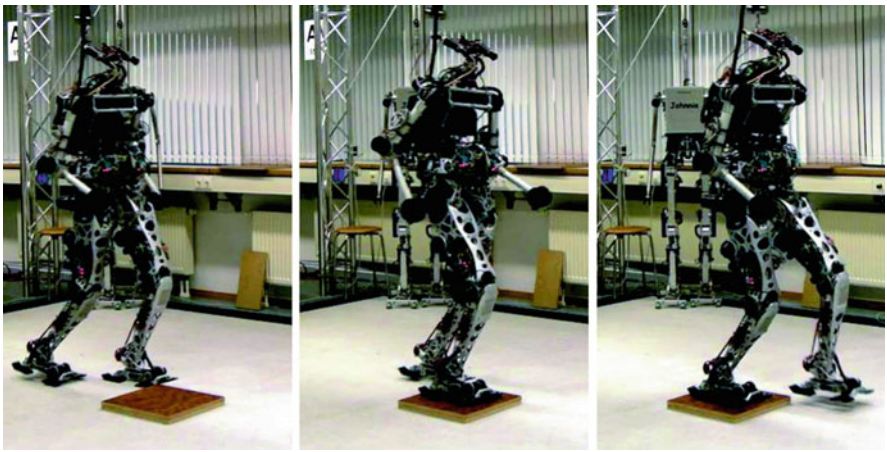


**Fig. 10.9** Lola walking over a 4 cm high board. There is no internal representation of the obstacle, Lola reactively replans her steps after detecting the ground using force sensing

## 10.6 Conclusion

This paper gave an overview of the control methods developed for and implemented on the biped robot Lola. The walking controller is implemented as a hierarchical system running on the onboard computer and local controllers. We have adopted a model-based approach to walking control and demonstrated the applicability of the proposed methods with the robots Lola and Johnnie. Real-time planning and stabilizing control allow the robot to quickly react to changing environments by replanning its steps, to resist external disturbances such as uneven terrain and to achieve fast and stable walking.

**Acknowledgements** This work is supported by the "Deutsche Forschungsgemeinschaft" (grant UL-105/29 and UL-105/34-1).

## References

1. Hirose M, Ogawa K (2006) Honda humanoid robots development. Phil Trans R Soc A 365:11–19
2. Tajima R, Honda D, Suga K (2009) Fast running experiments involving a humanoid robot. In: IEEE international conference on robotics and automation, ICRA '09, Kobe, Japan, May 2009, pp 1571–1576
3. BostonDynamics (2012/02/01) Company website for the petman robot. Online available: http://www.bostondynamics.com/robotpetman.html
4. Kaneko K, Kanehiro F,Kajita S, Hirukawa H, Kawasaki T, Hirata M,Akachi K, Isozumi T (2004) Humanoid robot hrp-2. In: Proceedings ICRA '04. 2004 I.E. international conference on robotics and automation, New Orleans, vol 2, Apr 2004, pp 1083–1090
5. Kaneko K, Harada K, Kanehiro F, Miyamori G, Akachi K (2008) Humanoid robot hrp-3. In: IEEE/RSJ international conference on intelligent robots and systems, IROS, Nice, France, Sep 2008, pp 2471–2478

6. Kaneko K, Kanehiro F, Morisawa M,Miura K, Nakaoka S, Kajita S (2009) Cybernetic human hrp-4c. In: Proceedings of the 9th IEEE-RAS international conference on humanoid robots, Humanoids, Paris, France, Dec 2009, pp 7–14
7. Ogura Y, Aikawa H, Shimomura K, Morishima A, ok Lim H, Takanishi A (2006) Development of a new humanoid robot wabian-2. In: Proceedings of IEEE international conference on robotics and automation, ICRA, Orlando, USA, May 2006, pp 76–81
8. Kim J-Y, Park I-W,Lee J, Kim M-S, Cho B-K, Oh J-H (2005) System design and dynamic walking of humanoid robot KHR-2. In: Proceedings of IEEE international conference on robotics and automation (ICRA), Barcelona, Spain, pp 1443–1448
9. Takenaka T, Matsumoto T, Yoshiike T (2009) Real time motion generation and control for biped robot –1st report: walking gait pattern generation–. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS), St. Louis, USA, Oct 2009, pp 1084–1091
10. Takenaka T, Matsumoto T, Yoshiike T, Shirokura S (2009) Real time motion generation and control for biped robot –2nd report: running gait pat tern generation–. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS), St. Louis, USA, Oct 2009, pp 1092–1099
11. Takenaka T, Matsumoto T, Yoshiike T, (2009) Real time motion generation and control for biped robot –3rd report: dynamics error compensation–. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS), St. Louis, USA, Oct 2009, pp 1594–1600
12. 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 IEEE/RSJ international conference intelligent robots and systems (IROS), St. Louis, USA, Oct 2009, pp 1601–1608
13. Website der honda motor corporation, new asimo, 2011. Online available: http://asimo.honda.com/asimotv/
14. Buschmann T, Schwienbacher M, Favot V, Ewald A, Ulbrich H (2012) The biped walking robot lola hardware design and walking control. Journal of the Robotics Society of Japan, (to appear)
15. Lohmeier S, Buschmann T, Ulbrich H (2009) System design and control of anthropomorphic walking robot LOLA. IEEE/ASME Trans Mechatron 14(6):658–666
16. Buschmann T, Favot V, Lohmeier S, Schwienbacher M, Ulbrich H, (2011) Experiments in fast biped walking. In: IEEE international conference on mechatronics (ICM), Istanbul, Turkey, Apr 2011, pp 863–868
17. Lohmeier S (2010) Design and realization of a performance enhanced humanoid robot. Ph.D. dissertation, Technische Universität München. Online available: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20101126-980754-1-4
18. Buschmann T (2010) Simulation and control of biped walking robots. Ph.D. dissertation, Technische Universität München. Online available: http://nbnresolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20101201-997204-1-6
19. Pfeiffer F, Löffler K, Gienger M, Ulbrich H (2004) Sensor and control aspects of biped robot "Johnnie". Int J Hum Robot (IJHR) 1(3):481–496
20. Buschmann T, Lohmeier S, Schwienbacher M, Favot V, Ulbrich H, Hundelshausen FV, Rohe G, Wuensche H-J (2010) Walking in unknown environments– a step towards more autonomy. In: Proceedings of IEEE-RAS international conference on humanoid robotics (humanoids), Nashville, USA
21. Buschmann T, Lohmeier S, Bachmayer M, Ulbrich H, Pfeiffer F (2007) A collocation method for real-time walking pattern generation. In: Proceedings of the 7th IEEE-RAS international conference on humanoid robots, Pittsburgh, USA, pp 1–6
22. Russel R, Shampine L (1972) A collocation method for boundary value problems. Numer Math 19:1–28, Springer Verlag
23. Fujimoto Y, Kawamura A (1996) Proposal of biped walking control based on robust hybrid position/force control. In: Proceedings of IEEE international conference on robotics and automation, Minneapolis, USA. vol 3, Apr 1996, pp 2724–2730
24. Buschmann T, Lohmeier S, Ulbrich H (2009) Biped walking control based on hybrid position/force control. In: IEEE/RSJ international conference on intelligent robots and systems, 2009. IROS 2009, St. Louis, USA, Oct 2009, pp 3019–3024

# Chapter 11
# Automated Kinematics Reasoning for Wheeled Mobile Robots

**Michael Hofbaur, Christoph Gruber, and Mathias Brandstötter**

**Abstract** Control schemes for wheeled mobile robots typically assume a specific mobility capability of a drive and implicitly use the drive's kinematics within its control procedures. This makes it difficult to deal with faults in the drive and to handle drives with diverse geometry and functionality that might even change during operation of a robot. As a consequence, we propose a model-based control scheme that builds upon an automated analysis of a robotic drive and on an on-line deduction of the drive's kinematics. We achieve this functionality through (1) the introduction of steering-angle independent, generalized variants of the rolling and sliding constraints for wheeled mobile robots and (2) the corresponding reformulation of kinematic analysis. This leads to a computationally efficient algorithm that deduces the (inverse) kinematics of a drive for its mode of operation or failure. Fault tolerant and robust behavior, however, is only one aspect of our control architecture. On-line kinematics analysis enables us to easily handle robots that change in geometry or functionality such as self-configuring modular robot systems and teams of cooperative robots.

## 11.1 Introduction

Wheeled mobile robots utilize a variety of application specific drives with different geometries and functionalities such as differential drives, drives with steered standard wheels or omni-directional drives with mecanum wheels, for example. This implies drive-specific kinematic capabilities. A controller uses the drive's kinematics to translate the requested robot motion into the appropriate actuation of the drive's wheels in terms of their rotational velocity and, whenever applicable, their steering angles.

M. Hofbaur (✉) • C. Gruber • M. Brandstötter
UMIT, Institute of Automation and Control Engineering, E. Wallnöfer Center 1, Hall in Tirol 6060, Austria
e-mail: michael.hofbaur@umit.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_11, 

Once in operation, the control law and also higher-level control layers, such as the path-planner, will use the drive's (inverse) kinematics implicitly through the implemented control algorithms. It is thus often impossible for such controllers to adapt their control laws once the kinematics of the drive changes significantly. Such a situation can occur in the case of a fault in the drive, for example, an impaired steering actuator. Another situation would be due to a geometry or functional adaption of the drive, for example, a drive that widens its base to increase stability or decreases its wheel distance to move through confined spaces [4]. Another realistic scenarios with implications to a drive's kinematics could be the adaption of the robot's number of wheels to deal with varying payload situations [8] or multi-robot transport [2]. Of course, one can solve such operational situations through specifically designed controllers that can account for pre-defined operational modes. However, it would be desirable to handle such situations in a more general and generic way.

To overcome these difficulties we propose a *model-programmed* control scheme that integrates control-design and control-execution within one intelligent solver for control. More precisely, the controller performs kinematic analysis for the drive on the basis of a *drive-model* at run-time (e.g. whenever the drive changes in geometry and functionality). As a side effect, we obtain a controller that is always aware of the drive's kinematic capabilities and that can perform re-configurations in the drive to automatically compensate for or recover from fault situations and notify a higher-level controller to re-plan the desired robot path according to the currently valid kinematic capabilities of the drive.

The key component of our control architecture is a *kinematics reasoning engine* that operates within the stringent time-constraints of real-time drive control. It thus implements the kinematic analysis and control design procedures as computationally efficient algorithm.

### *11.1.1 Background and Related Research*

Kinematics in robot drives is a well-studied field. A detailed analysis of the underlying mechanism can be found in text-books (e.g. [10]) and various publications (e.g. [1, 3]). In the latter reference, Campion et al. provide an unified framework that classifies robot drives according to their degree of mobility and degree of steerability and provides general procedures for the analysis of the robot-kinematics. However, almost all papers describe analysis and design procedures that are applied at the design stage of a robot and thus, assume functionally reasonable drives with dedicated topology. We intend to extend this scope to handle faults in the drive and drives with adaptive geometry/functionality. Faults, in particular, change the kinematics significantly and it is easily possible that one ends up with a degenerated robot with impaired mobility. Nevertheless it is important for an autonomous robot to cope with such situations as well. One can also find applications of robots with varying geometry or dynamically adaptive

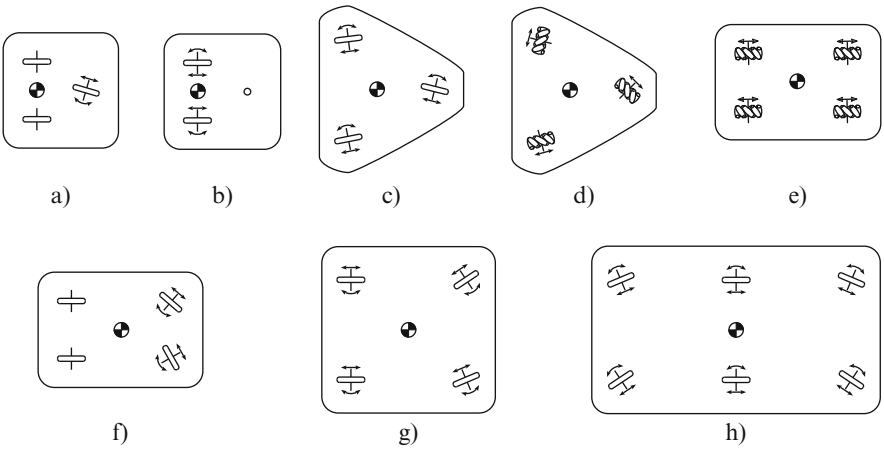



**Fig. 11.1** Wheeled mobile robot examples

number of wheels (e.g. see [4, 8]). However, these systems typically apply specialized drive controllers that handle few operational conditions. Our approach goes beyond this scope. We intend to provide a generic model-programmed control scheme in the spirit of [11] that can be applied to a wide spectrum of robot drives and thus, automatically include special situations such as anticipated and non-anticipated faults, re-configurable drives and multi-robot control scenarios.

## 11.2 Preliminaries

Wheeled mobile robots (WMR) provide mobility and load capabilities through various wheel arrangements as shown in Fig. 11.1. Many approaches, documented in literature, e.g. [3, 7], classify robots in terms of the wheels (steered-, unsteered-, mecanum-wheels) and their geometric arrangements and derive the kinematics accordingly. The basis for the robot's kinematics are the so called *rolling* and *sliding* conditions for the robot's wheels (e.g. see [3]) that we summarize in the following:

Focusing on the movement of the robot and neglecting the robot's pose within a global reference frame we can directly consider the robot with respect to a robot-body fixed reference frame $\Sigma_R : \{O_R; x_R, y_R\}$ . The robot's velocity with respect to an inertial frame expressed in the this robot-body frame $\Sigma_R$ is a vector

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \tag{11.1}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal velocities and $\dot{\theta}$ denotes the angular velocity.

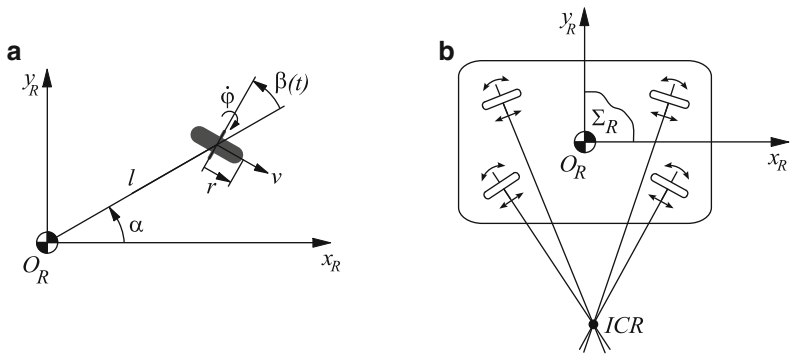



**Fig. 11.2** Robot wheel geometry and body-fixed reference frame $\Sigma_R$

Let us define the drive through its model that encodes the geometric alignment of the robot's wheels. We will use $\mathcal{W}$ to denote the set of $n$ wheels in a drive ($n = |\mathcal{W}|$) and specify the wheel alignment in terms of the wheel contact point expressed by the polar coordinates $l_i$ and $\alpha_i$ relative to $\Sigma_R$ and the wheel radius $r_i$ as shown in Fig. 11.2a for a wheel. The kinematics of a drive defines the possible movements for a drive and relates $\dot{\xi}$ to the rotational velocities $\dot{\varphi}_i$ and steering angles $\beta_i$ of the wheels $W_i \in \mathcal{W}$. The basis for a drive's kinematics is given through the *rolling* and *sliding* constraints. We introduce them for a centered orientable (steered) standard wheel $W_i$. The rolling constraint

$$r_i\,\dot{\varphi}_i = \begin{bmatrix} \sin(\alpha_i+\beta_i) & -\cos(\alpha_i+\beta_i) & -l_i\,\cos(\beta_i) \end{bmatrix} \dot{\xi} \tag{11.2}$$

$$= \mathbf{j}_i^T(\beta_i)\dot{\xi} \tag{11.3}$$

encodes the velocity $v$ *perpendicular to the wheel axis* and relates the rotational velocity $\dot{\varphi}_i$ of the wheel with the robot's velocity $\dot{\xi}$. The sliding constraint

$$0 = \begin{bmatrix} \cos(\alpha_i+\beta_i) & \sin(\alpha_i+\beta_i) & l_i\sin(\beta_i) \end{bmatrix} \dot{\xi} \tag{11.4}$$

$$= \mathbf{c}_i^T(\beta_i)\dot{\xi} \tag{11.5}$$

assures that there is no motion *perpendicular the wheel plane*. The rolling and sliding constraints of fixed oriented standard wheels are conceptually the same, except that $\beta_i$ is fixed and not time-dependent.

To derive the rolling and sliding constraints for the complete drive, one combines the individual row vectors $\mathbf{j}_i^T$ and $\mathbf{c}_i^T$ to form the matrices

$$\mathbf{J}_1(\beta) = \begin{bmatrix} \mathbf{j}_1^T(\beta_1) \\ \vdots \\ \mathbf{j}_n^T(\beta_n) \end{bmatrix}, \ \mathbf{C}_1(\beta) = \begin{bmatrix} \mathbf{c}_1^T(\beta_1) \\ \vdots \\ \mathbf{c}_n^T(\beta_n) \end{bmatrix}. \tag{11.6}$$

This leads to the overall rolling and sliding constraints for the robot

$$\mathbf{J}_1(\beta)\dot{\xi} = \mathbf{J}_2\dot{\varphi} \tag{11.7}$$

$$\mathbf{C}_1(\beta)\dot{\xi} = \mathbf{0}, \tag{11.8}$$

where $\mathbf{J}_2 = \text{diag}([r_i])$ combines the wheel radii $r_i$ into a constant diagonal matrix.

Of course, we allow a mixture of wheels with $n_c$ centered orientable (steerable) standard wheels and $n_f$ fixed oriented standard wheels[1] so that $n = n_c + n_f$. In that sense, we can group wheels accordingly and obtain

$$\mathbf{J}_1(\beta_s) = \begin{bmatrix} \mathbf{J}_{1c}(\beta_s) \\ \mathbf{J}_{1f} \end{bmatrix}, \mathbf{C}_1(\beta_s) = \begin{bmatrix} \mathbf{C}_{1c}(\beta_s) \\ \mathbf{C}_{1f} \end{bmatrix}. \tag{11.9}$$

The vector $\beta_s = [\beta_1, \ldots, \beta_{n_c}]^T$ combines all (time varying) steering angle parameters $\beta_i(t)$ for the $n_c$ centered orientable wheels.

The sliding constraint $\mathbf{C}_1(\beta_s)\dot{\xi} = \mathbf{0}$ is satisfied if $\dot{\xi} \in \ker(\mathbf{C}_1(\beta_s))$. If we consider two centered orientable standard wheels, then $1 \leq \text{rank}\,(\mathbf{C}_1(\beta_s)) \leq 2$. If we add more wheels, then the rank of $\mathbf{C}_1(\beta_s)$ can increase up to 3, meaning that no motion might be possible! To avoid that, one has to choose the steering angles $\beta_s$ so that $\text{rank}(\mathbf{C}_1(\beta_s)) \leq 2$. This demand on the rank of $\mathbf{C}_1(\beta_s)$ corresponds to the existence of a common intersection of all wheel axles at a finite distance from the robot center for $\text{rank}(\mathbf{C}_1(\beta_s)) = 2$ or at infinity for $\text{rank}(\mathbf{C}_1(\beta_s)) = 1$. In a perfectly controlled and actuated robot, this intersection defines the *instantaneous center of rotation (ICR)* of the robot-body motion (see Fig. 11.2b).

It is the task of the robot's low-level *kinematics* controller to maintain the steering angles and rotational speeds synchronously such that all wheel axles intersect and the sliding and rolling constraints define the appropriate ICR for the desired motion $\dot{\xi}$. However, this will only work if the path-planner or any other high-level control authority restricts itself to motion set-points $\dot{\xi}$ that are possible according to the drive's kinematics.

[1] It is straight forward to include mecanum wheels as well. The associated details are given in the Appendix.

## 11.3 Kinematics Reasoning

Our goal is to provide a generally applicable mechanism for the low level controller so that it can handle a large variety of drives with different geometric wheel alignments and wheel types. I.e. we are seeking for a computationally efficient algorithm that derives the kinematics or better the inverse kinematics from a model of the drive.

**Definition 11.1.** We define the model of a robot drive in terms of the tuple $\langle \Sigma_R, \mathcal{W} \rangle$ that specifies the robot-body fixed reference frame and the set of wheel models $\mathcal{W} = \{W_1, \ldots, W_n\}$ . Each wheel model defines a tuple $W_i = \langle T_i, G_i, M_i \rangle$ where $T_i$ details the wheel type {standard-fixed, standard-steered, mecanum}, $G_i$ specifies the wheel's geometry within $\Sigma_R$ in terms of the polar coordinates $\{l_i, \alpha_i, \beta_i\}$ , the wheel radius $r_i$ and $\gamma_i$ for mecanum wheels. Finally, $M_i$ captures the set of operational and fault modes for a specific wheel, e.g. {actuated, non-actuated, blocked-steering, blocked-rotation, ...}.

We allow the model to evolve over time. This includes a dynamically changing set of wheels and adaptations in wheel geometry. Furthermore, in our control architecture we use an additional functional unit that estimates the mode of operation or failure for each wheel and its associated actuators. Mode-estimation and fault detection, however goes beyond the scope of this paper and we refer the interested reader to [9].

Given the drive model and a mode specification that captures the operational/fault condition for a drive at a specific time point, we are interested to analyze a drive beyond the classification in terms of its degree of mobility and degree of steerability. In fact, it is our goal to explicitly compute the *space B of admissible and controllable motions* that fully describes the mobility capabilities and drive characteristics of a wheeled mobile robot.

We use this explicit knowledge in two ways. Firstly, we provide this information to higher-level control so that a path-planner automatically takes the kinematic constraints of a drive into account. Secondly, we enable our low-level controller to check, whether a drive command from the path-planner is admissible for the drive in its current mode of operation/failure. This can be done by evaluating the condition:

$$\dot{\xi} \in B\,. \tag{11.10}$$

A violation of (11.10) indicates an incompatible drive command that would infringe the kinematic constraints and thus, cannot be executed as desired by the low level controller. We therefore propose a control architecture that extends the low-level drive controller with an additional *kinematics reasoning unit* which (a) analyzes the kinematics of a drive during on-line operation of the robot, (b) provides the inverse kinematics for the low level controller and (c) checks drive commands on the basis of the condition (11.10). An invalid drive command is

thus captured and the kinematics reasoning unit can (d) either stop the robot and notify the path-planner about the failure to execute a desired drive command or (e) compute a re-configuration that maintains the operation of the robot to some extend in that it overrules the path-planner and deduces an alternative set-point $\dot{\xi}^*$ within $B$.

The basis for this operation of the robot drive controller is not a hard wired inverse kinematics model that provides the basis to compute the set-points for the wheel's steering angles $\beta_i$ and angular velocities $\omega_i = \dot{\varphi}_i$ but a general procedure that derives the inverse kinematics from the drive geometry and its present operational/fault condition. We achieve this functionality through introducing a generalized form of the rolling and sliding conditions that we call *qualitative rolling and sliding constraints*.

### 11.3.1 Kinematic Analysis with Qualitative Rolling and Sliding Constraints

The key algorithm for the kinematics reasoning procedure is the deduction of $B$. The kinematics of robot drives are typically described in terms of the *rolling* and *sliding constraints* given above in (11.7) and (11.8) where the matrices $\mathbf{J}_1$ and $\mathbf{C}_1$ depend upon the (time-varying) steering angles $\beta_s = [\beta_1, \ldots, \beta_{n_s}]^T$. In our *kinematics reasoning unit*, however, we want to decide, whether a velocity command $\dot{\xi}$ is admissible and controllable or not – *independently* from the steering angles of the wheels in the drive. Therefore, we take a more detailed look at the meaning of steering angle-independency for rolling and sliding constraints exemplary for a drive with $n_c$ standard centered orientable (steered) wheels and $n_f$ fixed oriented standard wheels.

Admissible movements for a robot are well understood and documented in literature [3, 7, 10]. Given the sliding constraint

$$\mathbf{C}_1(\beta_s)\dot{\xi} = \mathbf{0} \tag{11.11}$$

we deduce the admissible movements through null-space analysis of $\mathbf{C}_1$, i.e.

$$\ker(\mathbf{C}_1(\beta_s)) \subseteq \mathbb{R}^3. \tag{11.12}$$

The typical approach to obtain a steering angle independent analysis is to focus on the robot's unsteered wheels, as one can always select appropriate steering angles $\beta_s$ so that the steered wheels do not impose additional constraints.[2] We generalize this notion and provide *qualitative constraints* that dynamically capture the kinematics of a drive at its specific operational condition. Each wheel $W_i \in \mathcal{W}$ contributes a mode-specific *qualitative sliding constraint* $\mathbf{c}_{q,i}^T$ for the combined matrix

[2] Of course, this only holds for steered wheels with unconstrained steering angles.

$$\mathbf{C}_q = \begin{bmatrix} \mathbf{c}_{q,1}^T \\ \vdots \\ \mathbf{c}_{q,n}^T \end{bmatrix} . \tag{11.13}$$

For example, a standard wheel with blocked steering (at angle $\beta_i$) contributes the usual sliding constraint

$$\mathbf{c}_{q,i}^T := [\cos(\alpha_i + \beta_i) \quad \sin(\alpha_i + \beta_i) \quad l_i \ \sin(\beta_i)] . \tag{11.14}$$

A wheel with blocked steering and blocked rotation provides an additional constraint perpendicular to the wheel's axis (i.e. in rolling direction) so that we use

$$\mathbf{c}_{q,i}^T := \begin{bmatrix} \cos(\alpha_i + \beta_i) & \sin(\alpha_i + \beta_i) & l_i \ \sin(\beta_i) \\ \cos(\alpha_i + \beta_i + \frac{\pi}{2}) & \sin(\alpha_i + \beta_i + \frac{\pi}{2}) & l_i \ \sin(\beta_i + \frac{\pi}{2}) \end{bmatrix} . \tag{11.15}$$

A wheel with blocked rotation and operational steering leads to an analogous, steering angle independent constraint

$$\mathbf{c}_{q,i}^T := \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & l_i \ \sin(0) \\ \cos(\alpha_i + \frac{\pi}{2}) & \sin(\alpha_i + \frac{\pi}{2}) & l_i \ \sin(\frac{\pi}{2}) \end{bmatrix} \tag{11.16}$$

that prevents all movements of the robot except a rotation at the wheel's contact point. A fully operational centered orientable steered wheel, on the other hand, does not impose any constraint as noted above. We express this fact by using a null-vector

$$\mathbf{c}_{q,i}^T := [0 \quad 0 \quad 0] \tag{11.17}$$

for the qualitative sliding constraint matrix $\mathbf{C}_q$.

Combining the mode dependent qualitative sliding constraints for all wheels $W_i, i = 1, \ldots, n$, provides $\mathbf{C}_q$ which generalizes $\mathbf{C}_{1f}$ that specifies the sliding constraints for fixed orientation wheels. This allows us to obtain the *space of admissible motions* $Z$ for a specific configuration and mode of operation/failure of the drive through the null-space analysis

$$Z = \ker(\mathbf{C}_q) \subseteq \mathbb{R}^3 . \tag{11.18}$$

We now intend to provide an analogous analysis for the *space S of controllable movements*, i.e. the velocities $\dot{\xi} \in S$ that one can actuate through the motorized wheels in the drive. The implication of wheel actuation, i.e. $\dot{\varphi}_i \neq 0$ is captured through the rolling constraint

$$\mathbf{J}_1(\beta_s)\dot{\xi} = \mathbf{J}_2\dot{\varphi} . \tag{11.19}$$

A null-space analysis of $\mathbf{J}_1(\beta_s)$

$$\dot{\bar{\xi}} \in \bar{S}_1\,(\beta_s) = \ker(\mathbf{J}_1(\beta_s))\,, \tag{11.20}$$

however, provides the *complement* $\bar{S}_1$ to the space of controllable movements.

Let us analyze this fact with one single wheel $W_i$ at geometric position $\langle l_i, \alpha_i \rangle$ as shown in Fig. 11.2a to capture this implication in detail. Recall, that the wheel defines the following row

$$\mathbf{j}_{1,i}^T(\beta_i) = [\sin(\alpha_i + \beta_i) - \cos(\alpha_i + \beta_i) - l_i \cos(\beta_i)] \tag{11.21}$$

in the matrix $\mathbf{J}_1(\beta_s)$ of the rolling constraint. The null-space of $\mathbf{j}_{1,i}^T(\beta_s)$ has dimension 2. This can be argued by simple linear algebra: Let $\dot{\bar{\xi}} \in \mathcal{X} = \mathbb{R}^3$. Then, (11.19) is a linear map $f(\dot{\bar{\xi}}) : \mathcal{X} \to \mathbb{R}$ in the single wheel case. Thus,

$$\dim(\ker(f)) + \dim(\mathrm{im}(f)) = \dim(\mathcal{X}) = 3 \tag{11.22}$$

holds. The question of interest is now: How does the steering angle $\beta_i$ influence the null-space of $\mathbf{j}_{1,i}^T(\beta_s)$? For this purpose, we re-write the rolling constraint in a factorized form that follows from the addition theorems of trigonometry

$$\mathbf{j}_{\beta,i}^T(\beta_i)\mathbf{j}_{q,i}^T\dot{\bar{\xi}} = r_1\,\dot{\varphi}_i \tag{11.23}$$

where

$$\mathbf{j}_{\beta,i}^T(\beta_i) = [\,\cos(\beta_\mathrm{i})\ \sin(\beta_\mathrm{i})\,] \tag{11.24}$$

and

$$\mathbf{j}_{q,i}^T = \begin{bmatrix} \sin(\alpha_\mathrm{i}) & -\cos(\alpha_\mathrm{i}) & -l_i \\ \cos(\alpha_\mathrm{i}) & \sin(\alpha_\mathrm{i}) & 0 \end{bmatrix}. \tag{11.25}$$

Note that, until now, we have not made any changes to the rolling constraint. As mentioned before are we interested in the following question: Which motion set-points $\dot{\bar{\xi}}$ are not controllable for *any* steering angle $\beta$? The answer is found in the factorized rolling constraint (11.23), which emphasizes:

$$\begin{aligned} \bar{S} = \ker(\mathbf{j}_{q,i}^T) &\subset \ker(\mathbf{j}_{\beta,i}^T(\beta_i)\,\mathbf{j}_{q,i}^T) \\ &\subset \ker(\mathbf{j}_{1,i}^T(\beta_i)) \\ &\subset \bar{S}_1(\beta_i)\,. \end{aligned} \tag{11.26}$$

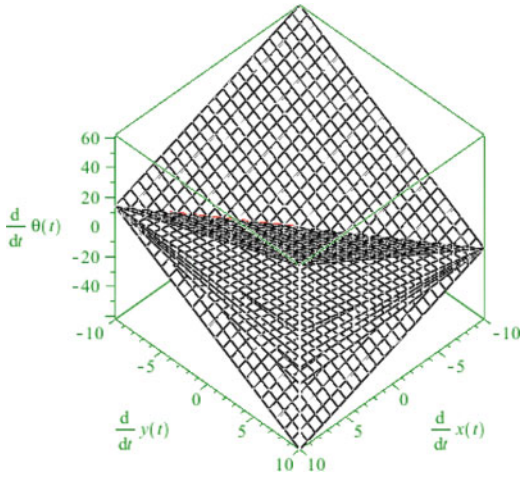



**Fig 11.3** Planes $\bar{S}_1(\beta)$ at different steering angles of one single wheel at $\alpha = \pi/4$ and $l = 1$. The rotation axis of the planes defines $\bar{S}$

Since $\mathbf{j}_{q,i}^T$ does not depend on $\beta_i$, its null-space $\ker(\mathbf{j}_{q,i}^T)$ delivers exactly the motion set-points,[3] which are non-controllable through wheel $W_i$ for any steering angle $\beta_i$. Note that $(\ker(\mathbf{j}_{q,i}^T)) = 1$. This means, that the space of qualitatively non-controllable velocities $\bar{S}$ has one dimension less than $\bar{S}_1(\beta_i)$. Thereby, the qualitative rolling constraint is less restricting than the effective rolling constraint, as it allows the steering angle to take the appropriate value. This corresponds to the apparent gain of a degree of freedom for the choice of controllable motion set-points.

If one thinks of $\mathcal{X}$ as three-dimensional space, the base vectors of $\bar{S}_1(\beta_i)$ span a plane (cf. 11.20). Any general velocity $\dot{\xi}$ can be depicted into components parallel to the plane and orthogonal to it. The parallel components violate the rolling constraint. Hence, all velocities having a non-zero component parallel to the plane are not controllable. Consistently, the set of controllable velocities is a straight line through the origin, which is orthogonal to the plane.

The set $\bar{S}$ has one degree of freedom – which corresponds to a straight line – and lies in $\bar{S}_1(\beta)$, independent of the steering angle $\beta$. Hence, with varying steering angle, the plane $\bar{S}_1(\beta)$ rotates around an axis $\bar{S}$ through the origin (see Fig. 11.3).

How can this result be interpreted, especially when there is more than one wheel? Again, we stack all qualitative rolling constraints to form $\mathbf{J}_q$

[3] More detailed analysis shows, that these are those motion set-points, which place the ICR in the wheel contact point.

$$\mathbf{J}_q = \begin{bmatrix} \mathbf{j}_{q,1}^T \\ \vdots \\ \mathbf{j}_{q,n}^T \end{bmatrix} \tag{11.27}$$

and analyze the null-space of $\mathbf{J}_q$ to obtain those motion commands, that cannot be actuated independently from the choice of the steering angles. However, this does not mean that velocity set-points that are not in this null-space can be driven! The non-controllable motions must be understood in a way similar to "non-controllable due to steering-angle independent under-actuation". Let us give the following examples: Consider a robot with:

1. One fixed standard wheel: $\ker(\mathbf{J}_q)$ will contain all motions that are not parallel to the rolling plane of the wheel.
2. One centered orientable standard wheel: $\ker(\mathbf{J}_q)$ will contain only those motions, that place the ICR in the contact point of the wheel.
3. One fixed standard wheel and one centered orientable wheel: $\ker(\mathbf{J}_q)$ will always be empty. The steered wheel is always able to turn the robot about the contact point of the fixed oriented wheel into the proper direction and execute a requested motion command.
4. Two centered orientable standard wheels: $\ker(\mathbf{J}_q)$ will always be empty.
5. Three fixed wheels, aligned in a way that they block any motion: $\ker(\mathbf{J}_q)$ will always be empty, meaning that all velocities are controllable – although no velocity can be driven! The disability of executing any motion will be handled by the sliding constraint.
6. Three fixed wheels, co-aligned in a way so that the rolling planes of the wheels are parallel: $\ker(\mathbf{J}_q)$ will contain all motions that are not parallel to the rolling planes of the wheels.

The specification of $\mathbf{j}_{q,i}^T$ in (11.4) describes a centered orientable standard wheel in the *nominal* case. A wheel with blocked steering simply acts as an unsteered wheel so that $\mathbf{j}_{q,i}^T$ becomes $\mathbf{j}_{q,i}^T = \mathbf{j}_{1,i}^T(\beta_i)$ . The situation for a drive with impaired rotational actuation (e.g. a freely spinning wheel or a wheel with blocked rotation) implies that the wheel $W_i$ cannot contribute to the robot's motorization. We express this fact through the null-vector $\mathbf{j}_{q,i}^T = [0 \;\; 0 \;\; 0]$. Table 11.1 summarizes the variants of $\mathbf{j}_{q,i}^T$ for operational and fault conditions together with the qualitative sliding constraints introduced above.

Summing up, our approach for kinematics reasoning proceeds as follows:

1. At the current time-point $t_k$, use the mode estimate $m(t_k)$ (provided through an estimation/diagnosis unit) and the currently valid drive's model $\mathcal{M}(t_k)$ to deduce the matrices of the rolling and sliding constraints $\{\mathbf{C}_q, \mathbf{J}_q, \mathbf{C}_1(\beta_s), \mathbf{J}_1(\beta_s), \mathbf{J}_2\}$.
2. Compute the *space of the admissible motions* $Z$ through

$$Z = \ker(\mathbf{C}_q) \subseteq \mathbb{R}^3 \,. \tag{11.28}$$

**Table 11.1** Qualitative rolling and sliding constraints for a standard wheel

| Mode of operation | $\mathbf{j}_q^T$ | $\mathbf{c}_q^T$ |
|---|---|---|
| OK: actuated rotation and operational steering | $\begin{bmatrix} \sin(\alpha) & \cos(\alpha) \\ -\cos(\alpha) & \sin(\alpha) \\ -l & 0 \end{bmatrix}^T$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ |
| Fault 1: actuated rotation and blocked steering | $\begin{bmatrix} \sin(\alpha+\beta) \\ -\cos(\alpha+\beta) \\ -l \cdot \cos\beta \end{bmatrix}^T$ | $\begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \\ l \cdot \sin\beta \end{bmatrix}^T$ |
| Fault 2: freely spinning wheel and blocked steering | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ | $\begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \\ l \cdot \sin\beta \end{bmatrix}^T$ |
| Fault 3: blocked rotation and operational steering | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ | $\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \\ 0 & l \end{bmatrix}^T$ |
| Fault 4: freely spinning wheel and operational steering | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ |

3. Compute the *space of non-controllable motions* $\bar{S}$ through

$$\bar{S} = \ker(\mathbf{J}_q) \subseteq \mathbb{R}^3 \,. \tag{11.29}$$

4. Whenever the two spaces intersect, i.e. $Z \cap \bar{S} \neq \emptyset$, we have to refine the admissible velocities $Z$ to exclude those movements that cannot be actuated through the robot's wheels. By computing the complement of $\bar{S}$

$$S = \ker(\bar{\mathbf{S}}^T) \subseteq \mathbb{R}^3 \,, \tag{11.30}$$

where $\bar{\mathbf{S}}$ denotes the *matrix of basis vectors* for $\bar{S}$, we obtain the *controllable* velocities so that, finally, the intersection

$$Z \cap S =: B \tag{11.31}$$

defines the space of *admissible and controllable* velocities for a given mode of operation or failure of the robot.

5. Validate a drive command through checking

$$\dot{\xi} \in B \,,$$

provide $\{\mathbf{C}_q, \mathbf{J}_q, Z, \bar{S}, S, B\}$ to higher-level control and compute the steering angles $\beta_s$ and rotational speeds $\dot{\varphi}$ for all steered and/or actuated wheels on the basis of $\{\mathbf{C}_1(\beta_s), \mathbf{J}_1(\beta_s), \mathbf{J}_2\}$.

The introduction of the *qualitative sliding* and, in particular, the *qualitative rolling constraints* enables us to re-formulate the well-known argument on null-space analysis of $\mathbf{C}_1$ (e.g. see [3, 10]) into an efficient algorithmic form that allows on-line drive-space and control/motorization analysis. The computations rely on several null-space computations (e.g. via singular-value decomposition) and the vector-space intersection (11.31) (e.g. with the Zassenhaus algorithm [12]). Illustrative examples for drive space computations can be found in [5]. All operations can be implemented efficiently so that one can utilize the reasoning concept directly within the drive's control loop and thus, reactively adapt the drive's control mechanism to the kinematics for the onset of operational modes and faults in the drive.

### *11.3.2 Applications*

Our kinematics reasoning capability provides adaptivity and a sense of *self-awareness* to the low level robot drive controller. I.e. it can validate and execute velocity set-points that are commanded through a higher-level controller for an adaptive robot drive. This is particularly important for autonomously dealing with faults in the drive.

For example, in [5] we demonstrated an efficient way to recover from a faulty steering actuator through geometric reasoning on the basis of the admissible and controllable velocity space $B$ . A blocked steering in an omni-directional robot leads to the loss of one-degree of freedom, so that we cannot choose the robot's velocity $\dot{\xi} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ arbitrarily. $B$ degenerates to a two-dimensional subspace, i.e. a plane as shown in Fig. 11.4. Using this knowledge, on can easily compute a feasible drive command, e.g. with adapted rotational speed $\dot{\Theta} \rightarrow \dot{\Theta}^*$ through projecting the desired velocity $\dot{\xi}$ onto the plane as shown in Fig. 11.4. This re-configuration procedure maintains the longitudinal velocities $\dot{x}$ and $\dot{y}$ and thus, maintains the robot's movement along the planned path.

Another aspect of self-awareness can be drawn from the qualitative rolling constraint matrices/vectors $\mathbf{j}_{q,i}^T$ . As we mentioned earlier, the null-spaces of $\mathbf{j}_{q,i}^T$ indicate critical velocities that put the ICR at wheel contact points. They capture the *singularities* of the drive. Providing this information to the path-planner, it is possible to actively avoid these critical velocities during operation of the drive. Furthermore, we were able to show that the qualitative rolling constraint matrices ($\mathbf{j}_{q,i}^T$) provide valuable information for robot-drive diagnosis as they allow one to formulate a kinematics-aware diagnosis and mode estimation procedure [9].

We use our control concept to deal with an onset of drive configurations which we can form with our modular robot platform [6] that allows us to configure

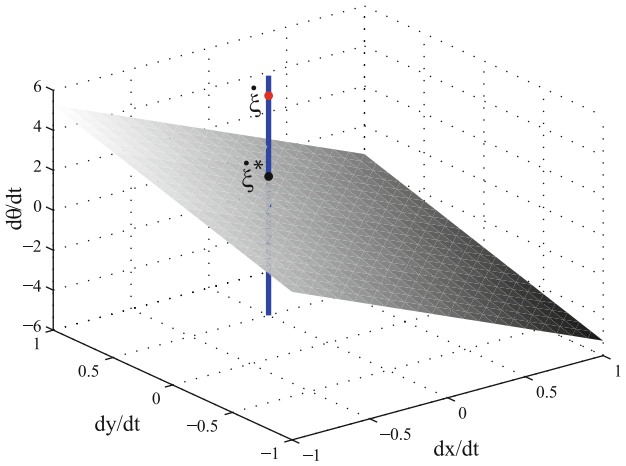



**Fig. 11.4** Drive command adaption $\dot{\xi} \rightarrow \dot{\xi}^*$ to maintain the longitudinal velocities $\dot{x}, \dot{y}$ (cf. [5])

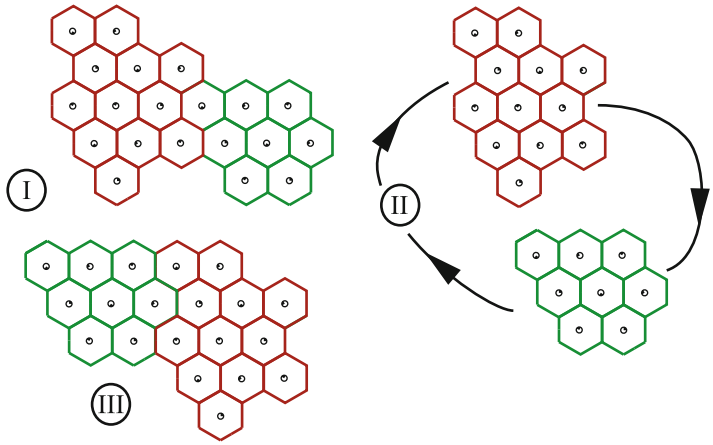



**Fig. 11.5** Reconfigurable robot drive built from hexagonal drive modules

a large variety of robot drives and, in particular, re-configurable robots from hexagonal drive modules. Figure 11.5 shows such a compound robot-drive and its reconfiguration schematically. Controlling such an adaptive drive with a potentially large number of wheels is surely non-trivial. Our model-based approach with

kinematic reasoning as its key component, however, allows us to deal with time-variant robot topologies and functionalities so that we can straightforwardly handle such complex robot-drives.

## 11.4 Conclusions and Outlook

Traditional kinematics analysis techniques are well suited for off-line robot drive design. Our intention, however, was to build a generic robot controller that can be applied to a wide variety of robot drives. Therefore, we presented a scheme for automated reasoning that analyzes and deduces the kinematics of a robot drive. Our algorithmic solution builds upon qualitative sliding and rolling constraints. These constraints represent steering-angle independent constraints that allow us to efficiently analyze a robot drive in terms of its admissible and controllable movements as well as its singularities. An efficient algorithmic formulation enables us to perform kinematics reasoning within the control-loop during the run-time of the robot. As a consequence, we can directly deal with drives that adapt in terms of geometry and functionality, as well as with typical faults that, otherwise, significantly change the kinematic behavior of the drive. Performing kinematics reasoning within the controller of the robot drive leads to a sense of self-awareness. This enables interesting new perspectives for robot control, such as intelligent, kinematics-aware path planning and alternative approaches for re-configurable robots and coordinated control of multi-robot systems.

To achieve these goals, we are currently working on a re-formulation of kinematics reasoning as distributed algorithm. This functionality, together with an autonomous modeling capability will enable us to deal with complex robot structures even more easily. Another line of research is the integration of kinematics reasoning with higher-level drive control. For example, we intend to use the drive's self-awareness to deduce appropriate mobility classifications that enables a hybrid control algorithm to schedule, or even deduce on-line, the appropriate control strategy to maintain an autonomous and robust behavior of a robot.

**Acknowledgement** This work has been funded through the Austrian Science Fund (FWF) under grant no. P20041-N15.

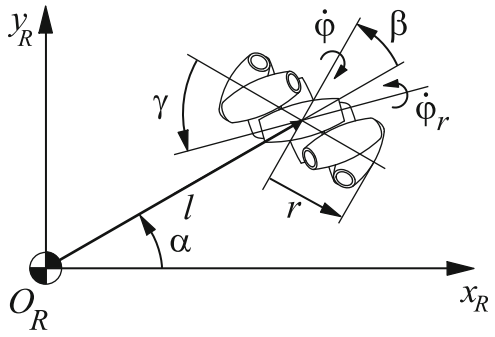



**Fig. 11.6** Mecanum wheel geometry

## Appendix

### *Qualitative Rolling and Sliding Constraints for an Omni-Directional Mecanum Wheel*

It is straight forward to provide qualitative sliding constraints for Mecanum wheels as well. If one considers a mecanum wheel with the geometry as shown in Fig. 11.6, one obtains the standard rolling and sliding constraints as

$$\begin{aligned} r\dot{\varphi}\cos(\gamma) &= [\sin(\alpha+\beta+\gamma) \; -\cos(\alpha+\beta+\gamma) \quad -l\cos(\beta+\gamma)\,]\dot{\xi} \\ &= \mathbf{j}^T\dot{\xi}\,, \end{aligned}$$

$$\begin{aligned} 0 &= [\,\cos(\alpha+\beta+\gamma) \quad \sin(\alpha+\beta+\gamma) \quad l\sin(\beta+\gamma)\,]\,\dot{\xi} - r\dot{\varphi}\,\sin(\gamma) - r_r\,\dot{\varphi}_r \\ &= \mathbf{c}^T\dot{\xi}\,. \end{aligned}$$

Note that a fully operational mecanum wheel (actuated or freely spinning) does not impose any constraints on the robot's movement as the rotation of the individual rolls ensures the sliding constraint 1.32. We express this fact in terms of a qualitative constraint with $\mathbf{c}_q^T := [\,0 \quad 0 \quad 0\,]$ . The rolling constraint is time-invariant as mecanum wheels are typically used without active steering. Therefore, we obtain $\mathbf{j}_q^T := [\,\sin(\alpha+\beta+\gamma) \quad -\cos(\alpha+\beta+\gamma) \quad -l\cos(\beta+\gamma)\,]$ . However, a blocked mecanum wheel exhibits a behavior, where the individual rolls act like non-actuated standard wheels at $\beta+\gamma+\frac{\pi}{2}$ . The according qualitative sliding constraint is thus

$$\mathbf{c}_q^T := \left[\,\cos(\alpha+\beta+\gamma+\tfrac{\pi}{2}) \quad \sin(\alpha+\beta+\gamma+\tfrac{\pi}{2}) \quad l\sin(\beta+\gamma+\tfrac{\pi}{2})\,\right] .$$

We summarize the matrix entries $\mathbf{c}_q^T$ and $\mathbf{j}_q^T$ for the qualitative constraint matrices $\mathbf{C}_q$ and $\mathbf{J}_q$ in Table 11.2.

**Table 11.2** Qualitative rolling and sliding constraints for an omni-directional mecanum wheel

| Mode of operation | $\mathbf{j_q^T}$ | $\mathbf{c_q^T}$ |
|---|---|---|
| OK: actuated rotation | $\begin{bmatrix} \sin(\alpha+\beta+\gamma) \\ -\cos(\alpha+\beta+\gamma) \\ -l\cdot\cos(\beta+\gamma) \end{bmatrix}^T$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ |
| Fault 1: blocked rotation | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ | $\begin{bmatrix} \cos\left(\alpha+\beta+\gamma+\frac{\pi}{2}\right) \\ \sin\left(\alpha+\beta+\gamma+\frac{\pi}{2}\right) \\ l\cdot\sin\left(\beta+\gamma+\frac{\pi}{2}\right) \end{bmatrix}^T$ |
| Fault 2: freely spinning | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ | $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$ |

## References


1. Alexander J, Maddocks J (1989) On the kinematics of wheeled mobile robots. Int J Robot Res 8:15–27
2. Bhatt R, Tang C, Krovi V (2009) Formation optimization for a fleet of wheeled mobile robots – a geometric approach. Robot Auton Syst 57:102–120
3. Campion G, Bastin G, D'Andréa-Novel B (1996) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. IEEE Trans Robot Autom 12:47–62
4. Robuffo-Giordano P, Fuchs M, Albu-Schäffer A, Hirzinger G (2009) On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs. In: Proceedings of the IEEE international conference on robotics and automation (ICRA09), pp 4080–4087
5. Hofbaur M, Brandstötter M, Schörghuber C, Steinbauer G (2010) On-line kinematics reasoning for reconfigurable robot drives. In: Proceedings of the IEEE international conference on robotics and automation (ICRA10), pp 5441–5446
6. Hofbaur M, Brandstötter M, Jantscher S, Schörghuber C (2010) Modular reconfigurable robot drives. In: Proceedings of the international conference on robotics and automation and mechatronics (RAM 2010), pp 150–155
7. Kim W, Lee S, Yi B (2002) Mobility analysis of planar mobile robots. In: Proceedings of the IEEE international conference on robotics and automation (ICRA02), pp 2861–2867
8. Nüchter E, Bahr E, Lohmüller H (2008) Omnidirektionales Fahrzeug, Fahrmodul und mobiler Industrieroboter. German Patent Office, Offenlegungsschrift DE 10 2007 016 662 A1
9. Rienmüller T, Gruber C, Brandstötter M, Hofbaur M (2011) Odometry-based fault diagnosis for pseudo-omnidirectional wheeled mobile robots. In: Proceedings of the 22nd international-workshop on principles of diagnosis (DX11), Murnau, Germany, pp 29–35
10. Siegwart R, Nourbakhsh I (2004) Introduction to autonomous mobile robots. MIT Press, Cambridge
11. Williams B, Ingham M, Chung S, Elliott P, Hofbaur M, Sullivan G (2003) Model-based programming of fault-aware systems. AI Mag 24(4):61–75
12. Zassenhaus H (1948) Über einen Algorithmus zur Bestimmung der Raumgruppen. Commentarii Mathematici Helvetici 21:117–141

# Chapter 12
# Automatic Parameter Identification for Mechatronic Systems

**Rafael Ludwig and Johannes Gerstmayr**

**Abstract** Simulations of the behaviour of complex mechatronic systems require optimal simulation parameters for obtaining realistic results. For highly accurate mechatronic simulations, an algorithm for searching the optimal parameters is required. In the field of robotics the identification based on minimization of the residuum with least square methods is state of the art. This chapter describes a special algorithm for automatic parameter identification for mechatronic systems, based on the theory of genetic optimization, which works also in case of multiple local minima of the simulation error distribution. Nominal parameters of a simulated belt drive are identified in time and frequency domain highly accurate. Special treatment of the simulation error in frequency domain leads to reduced identification effort. Finally, the algorithm for automatic parameter identification searches real robot parameters up to high accuracy. The automatic parameter identification algorithm leads to accurate simulation results, even though the measurement contains noise and also time delays.

## 12.1 Overview

Mechatronic systems like robots have high demands for speed and accurate positioning. Such mechatronic systems require detailed models of the components e.g. gear boxes, the controller and the elasticity and damping of mechanical parts.

The coupling of trajectory planning and parameterized simulation for robots and have been implemented in [12, 13]. In many cases, certain system parameters are unknown, due to a variety of component suppliers. In previous work [14], an identification method for minimization of the error in time domain is described. This chapter describes the algorithm for automatic parameter identification, which

R. Ludwig (✉) • J. Gerstmayr
Austrian Center of Competence in Mechatronics GmbH, Altenberger Street 69, Linz 4040, Austria
e-mail: rafael.ludwig@accm.co.at; johannes.gerstmayr@accm.co.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_12, 

operates in time and frequency domain. Numerical tests for suitability and the identification of real mechatronical system parameters are content of this chapter, where the excellent behavior of the automatic parameter identification is shown without using algebraic manipulation of the system equations.

In the presence of noise,[1] evolution in nature is very effective, see [6]. If more than one (local) minimum exists in the simulation error, also called cost function, Newton's method must be initialized near the global minimum to find the best parameters. The advantage of genetic algorithms is their independence of initial conditions. In literature, different genetic algorithms are used for path planning like in [4, 10, 18], optimization of the controller parameters, see [1], parameter estimation [17, 20, 21, 22], tracking [15] but also image processing, compare [6]. Furthermore, problems with limited possibilities of parameter combination exist, see [1], and algorithms with binary parameters, named genes.

A direct approach to parameter identification is to manipulate and transform the model equations into the Laplace domain like in [7, 8]. The algebraic effort increases with the complexity of the model equations. In order to get enough information about the behavior about the system, the system input must be chosen carefully, see [5, 11].

Genetic algorithms are very suitable for parameter estimation, especially if the model equations representing the mechatronic system, are not fully known. Therefore a comparison of many different models can be easily done without algebraic manipulations for the identification. In this chapter, the implementation and application of a special algorithm for automatic parameter identification of mechanical and controller parameters of a robot are described, i.e. searching the correct parameters of the mechatronic system in a bounded parameter space. The Discrete Fourier Transformation is useful for computation of the joint velocities based on postion signals and for application of ideal filters depending on the sample rate of measured drive torque, see [16].

In the present chapter the original data of the trajectory planning and the recorded torque of the electrical drives is used by the automatic parameter identification. The equations of motion of the robot are assembled in a successive way, using the method like in [2, 19]. The multibody code HOTINT, described in [9], allows the coupled simulation of flexible multibodies, sensors, actuators and controller circuits. Thus, the mechatronic models and the automatic parameter identification are implemented within this code. The mechatronic system needs to be given in parametric form in any kind of simulation environment, which could be MATLAB, DYMOLA or similar, as well. However, the proposed automatic parameter identification does not need to know the structures of the mechatronic systems. Our proposed identification is named automatic because the parameter identification does not need algebraic manipulation of the system equations of the mechatronical systems, so the mathematical models and the number of parameters and also the parameter names can be switched very easy by the user. The proposed

[1] Note: Noise is contained in measurement data of mechatronic systems.

automatic parameter identification uses a shooting method for the first parameter generation. A special algorithm based on the theory of genetic optimization generates parameters of the following generations. Newton's method is an alternative for computation of following generations, but it requires differentiable cost function distributions with respect to the parameters, which is not always given in real mechatronic systems.

The structure of the present paper is organized as follows. In Sect. 12.2, our proposed identification for mechatronic systems is described. Cost functions based on the L2-norm in time and frequency domain are described. As example for a mechatronic system, a very simple belt drive model with well known nominal parameters is shown in Sect. 12.3. In order to show the advantages of the use of weights and cost functions in frequency domain, the results of the automatic parameter identification using cost functions in time and frequency domain are contained in this section. The identification of the nominal parameters is difficult, because the parameter search intervals are chosen very large and the belt drive model is undamped. This causes very sharp peaks in the amplitude spectrum of the cost function in frequency domain. Our proposed algorithm is applied to the identification of drive parameters of a real robot in Sect. 12.4 based on recorded drive torque, i.e. containing measurement delay and noise.

## 12.2 Automatic Parameter Identification Algorithm

The algorithm for automatic parameter identification, in the following denoted as identification, searches the optimal model parameter vector $\boldsymbol{\theta}_{opt}$ in a systematic way. Especially if multiple minima appear in the cost function, see Sect. 12.2.1, the algorithm has a big advantage in comparison with Newton's method, where often only a local minimum is found near the initial conditions. The identification is easy to use and has only a few user defined options like the components of the parameter vector $\boldsymbol{\theta}_{opt}$, containing the parameters of the mechatronic system to optimize and the search limits of the components. In the parameter space $P$, the first generation $g = 1$ of parameters is uniformly distributed.

The mean quadratic output errors $e^{gc}$ , shortly $e$, result from simulation with the child parameter vectors $\boldsymbol{\theta}^{gc}$. The index $c$ represents the number of a child parameter vector of the generation $g$. In the first generation

$$c = (1, 2, ..., N_{first}), \tag{12.1}$$

with first $N_{first}$ initial values, while in further generations, $g > 1$, the children are out of the set

$$c = (1, 2, ..., N_c N_s). \tag{12.2}$$

Here, $N_s$ is the number of surviving parameters and $N_c$ is the number of children of each generation. The child parameter set of generation,

$$g = (1, 2, ..., N_g), \tag{12.3}$$

is defined as

$$C^g = \left(\boldsymbol{\theta}^{g1}, \boldsymbol{\theta}^{g2}, ..., \boldsymbol{\theta}^{gN_sN_c}\right), \tag{12.4}$$

where $N_g$ is the maximal number of generations.

The surviving parameter set

$$S^g = \left(\boldsymbol{\theta}^{g1}, \boldsymbol{\theta}^{g2}, ..., \boldsymbol{\theta}^{gN_s}\right) \tag{12.5}$$

contains the surviving parameter vectors $\boldsymbol{\theta}^{gs}$ with index

$$s = (1, 2, ..., N_s), \tag{12.6}$$

which are chosen during the selection process, see Sect. 12.2.3. The surviving parameter vectors $\boldsymbol{\theta}^{gs}$ are parents of the next generation of child parameter vectors, using the principle of mutation

$$\boldsymbol{\theta}^{gs} \rightarrow \boldsymbol{\theta}^{(g+1)c}, \tag{12.7}$$

i.e. described in Sect. 12.2.4.

The mean quadratic output errors of the next generations are computed in the time or frequency domain, see Sect. 12.2.1. After a maximal number of generations is computed, the algorithm for identification stops and the optimal model parameter vector $\boldsymbol{\theta}_{opt}$ with the lowest cost function results.

### *12.2.1 Computation of Cost Function*

The $M$-dimensional measured system output vector $\boldsymbol{y}_{mes}(t_j)$ depends on discrete time points $t_j$, $j$=(1, 2, …, $N_t$). This vector is the base of the computation of the $M$-dimensional vector of mean quadratic simulation errors $\mathbf{e}$ depending on the child parameter vector $\boldsymbol{\theta}^{gc}$. By use of the corresponding vector of simulation outputs $\boldsymbol{y}_{sim}\left(\boldsymbol{\theta}^{gc}, t_j\right)$, the error vector results to

$$\boldsymbol{e}\left(\boldsymbol{\theta}^{gc}, t_j\right) = \boldsymbol{y}_{mes}(t_j) - \boldsymbol{y}_{sim}\left(\boldsymbol{\theta}^{gc}, t_j\right). \tag{12.8}$$

After each simulation, the cost function, i.e. the simulation error,

$$e^{gc} = \sum_{k=1}^{M} \sqrt{\frac{1}{N_t} \sum_{j=0}^{N_t-1} \left[e^k\left(\boldsymbol{\theta}^{gc}, t_j\right)\right]^2} \tag{12.9}$$

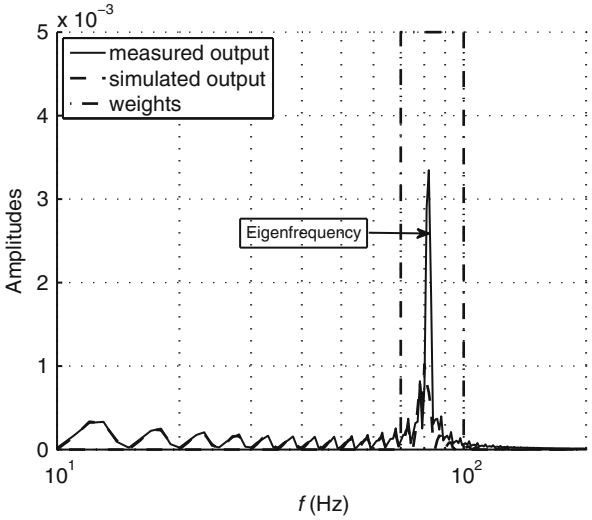



**Fig. 12.1** The difference of the measured and simulated output amplitude spectra leads to the error $\hat{E}$, which allows the application of ideal filters in form of weights in frequency domain

is evaluated and stored. The goal of the algorithm for identification is the search of the optimal parameter vector $\boldsymbol{\theta}_{opt}$ such that the cost function value is minimal. The number of optimized parameters $N$ is the dimension of the parameter space $P \subset \mathbb{R}^N$.

Due to the fact, that $\mathbb{R}^N$ is infinite and the identification process has to be finished in finite time, the components $\theta_i^{gc}$ of the parameter vectors $\boldsymbol{\theta}^{gc}$ are limited by the user,

$$\boldsymbol{\theta}_i^{gc} = [\theta_{i,min},\ \theta_{i,max}]. \tag{12.10}$$

The Eigenfrequencies of mechatronic systems are very important. In order to give the cost function in specific frequency intervals more or less influence (e.g. near the Eigenfrequencies), the cost function is transferred in the frequency domain. Using Parseval's theorem for discrete Fourier Transform (DFT) and the symmetric characteristic of its output[2] (12.9) can be rewritten as

$$e^{gc} = \frac{1}{2} \sum_{k=1}^{M} \sqrt{\left|\hat{E}_0^k\left(\boldsymbol{\theta}^{gc}\right)\right|^2 + 2 \sum_{j=1}^{\lfloor N_t/2 \rfloor} \left[\hat{E}_j^k\right]^2}. \tag{12.11}$$

Note, that $\hat{E}_j^k$ are the amplitudes in the frequency domain, see Fig. 12.1. Each amplitude belongs to a specific frequency. The application of frequency dependent weight factors $w_k$ is obtained using the transformation $\hat{E}_j^k \rightarrow w_k \hat{E}_j^k$ and (12.11). The use of factors $w_k \neq 1$ leads to a modified distribution of the cost function values in the parameter space $P$.

[2] If the DFT is applied to signals with real values, the result constists of conjugate complex values.

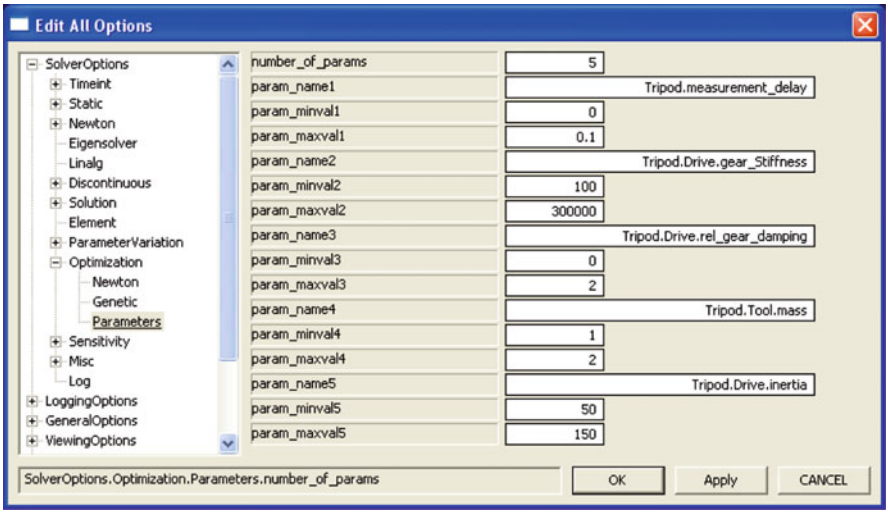



**Fig. 12.2** Initialization of the parameters of the automatic parameter identification, e.g. names, limits and number of parameters are very easy to handle

### *12.2.2 Initialization*

The user defined number of uniform distributed parameters of the first generation of the identification is called initial population size. The likelihood to find the global minimum, but also the computational effort depends on the initial population size. During the initialization, the user defines certain optimization settings, e.g. limits and number of parameters, see Fig. 12.2.

### *12.2.3 Selection Process*

After the cost function values $e^{gc}$ are computed with (12.9) or (12.11), the set of surviving parameters $S^g$ is defined, which contains parameter vectors $\boldsymbol{\theta}^{gc}$ with the lowest cost function values $e^{gc}$. The majority $|S^g|$ of this set is defined by the user. In order to keep the best parameters in generations $(g>1)$, the survivors of the previous generation are also considered during the selection process.

### *12.2.4 Mutation*

The values of the mutated parameters, also called child parameters, are located close to their parent parameters, i.e. the surviving parameters. The first generation is uniform distributed

$$\theta_i^{gc} = \theta_{i,min} + \left(\theta_{i,max} - \theta_{i,min}\right) r_i^{gc},\ r_i^{gc} \in [0,1], \tag{12.12}$$

with new random values $r_i^{gc}$ for every component $\theta_i^{gc}$ of each parameter vector. Related to the inverse formula of standard normal distribution, the function

$$s(x) = \begin{cases} \operatorname{sgn}\left(x - \frac{1}{2}\right)\sqrt{-\log\left(2\left|x - \frac{1}{2}\right|\right)} & \text{if } 0 \le x < \frac{1}{2} \text{ or } \frac{1}{2} < x \le x \\ 0 & \text{if } x = \frac{1}{2} \end{cases} \tag{12.13}$$

is used for parameter mutation. The mutated parameters $\boldsymbol{\theta}^{(g+1)c}$ for the next generation $g+1$ are computed with the user defined range reduction factor $\xi \in ]0,1]$. This factor is smaller than one in order to have decreasing influence to the distance between the child parameters and the surviving parameters

$$\theta_i^{(g+1)c} = \theta_i^{gs} + \left(\theta_{i,max} - \theta_{i,min}\right) s\left(r_i^{gc}\right) \xi^{(g-1)}. \tag{12.14}$$

After (12.14), each child parameter $\theta_i^{(g+1)c}$ is compared with its limits $\theta_{i,min}$ and $\theta_{i,max}$. In case of an exceeded limit, the child parameter $\theta_i^{(g+1)c}$ is set to this limit. In case of multiple minima of the cost function in $P$, the identification possibly searches with too many child parameters near the wrong minima. Therefore, a modified strategy of the selection process is shown in Sect. 12.2.5.

### 12.2.5 Modified Strategy for Selection Process

In order to be able to take influence on the selection process, an extended strategy for dealing with more than one multiple (local) minima of the cost function is showed in this section. Since the identification possibly searches near a local instead of the global minima of the simulation error in $P$. The identification should find several local minima at different locations in parameter space, also if the parameters with lowest cost function of the actual generation are located very close together. Therefore, the strategy of 12.2.4 is modified.

After all parameters of the generation are sorted with respect to their cost functions, the parameter vector leading to the lowest cost function is added to the set $S^g$. All other sorted parameters $\boldsymbol{\theta}^{gc} \in C^g$ have a distance $d(\boldsymbol{\theta}^{gc})$, which has to be bigger than a minimal allowed distance $d_{min,0}\xi^{(g-1)}$ to all parameters in $S^g$,

$$d(\boldsymbol{\theta}^{gc}) = \min_{\boldsymbol{\theta}^{gs} \in S^g} \left( \sqrt{\sum_i \left( \frac{\theta_i^{gc} - \theta_i^{gs}}{\theta_{i,max} - \theta_{i,min}} \right)^2} \right) > d_{min,0}\xi^{(g-1)}. \tag{12.15}$$

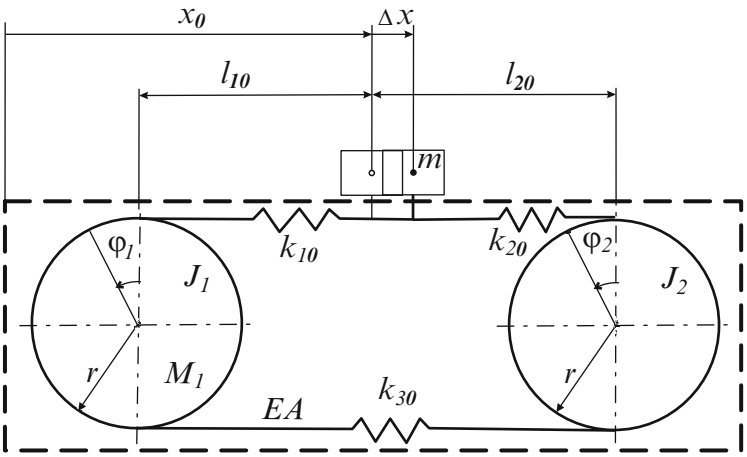



**Fig. 12.3** Belt drive model with stiffnesses, driving gear (index 1) and driven gear (index 2), axial belt stiffness $EA$ and spring lengths $l_{i0}$

Note, that the minimal allowed distance is decreasing every generation $g$, so it is possible to find minima of the cost function, which are located very close to other minima during the identification. The case of one dimensional parameter identification, the minimal distance factor has to be chosen

$$d_{min,0} < 1, \tag{12.16}$$

for good results of the identification. In $N$-dimensional parameter spaces, the minimal distance factor should be also chosen smaller than the diagonal of normed parameter space,

$$d_{min,0} < \sqrt{N}. \tag{12.17}$$

## 12.3 Application of Identification at Belt Drive

For a comparison of the results of the identification with cost function in time (compare [14]) and frequency domain, a belt drive simulation with exactly known parameters (axial stiffness model is according to [3]) is investigated, see Fig. 12.3. The initial angle of the driving gear $\varphi_1$ and the angle of the driven gear $\varphi_2$ are zero. According to [23] the assumption of small axial strains of the belt and a linear distribution of the displacement with respect to the initial belt lengths $l_i(x_0)$ is applied

$$l_i(x_0) = l_{i0}. \tag{12.18}$$

Following terms are assumed to be small

$$\begin{aligned} \Delta x &\ll l_{i0}, \\ \varphi_1 r &\ll l_{i0}, \\ \varphi_2 r &\ll l_{i0}. \end{aligned} \tag{12.19}$$

The displacements between initial and actual belt lengths depend on the rail displacement $\Delta x$, the angle of the driving gear $\varphi_1$ and the driven gear $\varphi_2$ yield

$$\begin{aligned} \Delta u^1 &= r\varphi_1 + \Delta x, \\ \Delta u^2 &= -r\varphi_2 - \Delta x, \\ \Delta u^3 &= r(\varphi_2 - \varphi_1). \end{aligned} \tag{12.20}$$

The stiffnesses of the belt result to

$$k_{i0} = \frac{EA}{l_{i0}} \forall i \in \{1, 2\}, k_{30} = \frac{EA}{l_{10} + l_{20}}. \tag{12.21}$$

The Lagrange equations of motion of the system can be written as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} = \frac{\partial L}{\partial q_j} + Q_j, \quad q_j \in \{\varphi_1, \varphi_2, x\}, \tag{12.22}$$

in which the Lagrangian

$$L = \frac{1}{2} \left( m \dot{x}^2 + J_1 \dot{\varphi}_1^2 + J_2 \dot{\varphi}_2^2 - \sum_{i=1}^{3} k_{i0} \Delta u^{i2} \right) \tag{12.23}$$

consists of the potential energy of the belt depending on $\Delta u^i$ from (12.20), and the quadratic terms of the kinetic energy of the gear inertias $J_i$ and the rail mass $m$. The principle of virtual work applied to the driving torque $M_1$ leads to the generalized forces

$$\delta W = M_1 \delta \varphi_1 \Rightarrow Q_1 = M_1. \tag{12.24}$$

The equations of motion follow to

$$\begin{aligned} J_1 \ddot{\varphi}_1 &= r[k_{30} r(\varphi_2 - \varphi_1) - k_{10}(r\varphi_1 + \Delta x)] + M_1, \\ J_2 \ddot{\varphi}_2 &= -r[k_{20}(r\varphi_2 + \Delta x) + k_{30} r(\varphi_2 - \varphi_1)], \\ m\ddot{x} &= -k_{20}(r\varphi_2 + \Delta x) - k_{10}(r\varphi_1 + \Delta x). \end{aligned} \tag{12.25}$$

In order to rotate the driving gear with a prescribed angle $\varphi_{1d}$, it is necessary to add a controller to the system. Therefore a $P$-controller is used to compute the driving torque

$$M_1 = P(\varphi_{1d} - \varphi_1). \tag{12.26}$$

**Table 12.1** Nominal parameters of belt drive simulation

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Initial spring length of $k_{10}$ | $l_{1,0}$ | 0.1875+$x_0$ | m |
| Initial spring length of $k_{20}$ | $l_{2,0}$ | 0.7875-$x_0$ | m |
| Axial stiffness | $EA$ | $2 \times 10^5$ | N |
| Moment of inertias | $J_1, J_2$ | $1 \times 10^{-1}$ | kg m$^2$ |
| Radius | $r$ | $1 \times 10^{-3}$ | m |
| Rail mass | $m$ | 0.948 | kg |
| Controller gain | $P$ | 45 | Nm rad$^{-1}$ |

The mass and stiffness matrices of the controlled system are

$$\mathbf{M} = \mathrm{diag}(J_1,\ J_2,\ J_3), \tag{12.27}$$

$$\mathbf{K} = \begin{pmatrix} k_{10}r^2 + k_{30}r^2 + P & -k_{30}r^2 & k_{10}r \\ -k_{30}r^2 & k_{20}r_2^2 + k_{30}r^2 & k_{20}r \\ k_{10}r & k_{20}r & k_{10} + k_{20} \end{pmatrix}. \tag{12.28}$$

The parameters for the nominal simulation are shown in Table 12.1. The Eigenfrequencies result to

$$\det(\mathbf{K} - \omega^2\mathbf{M}) = 0 \Rightarrow \omega_{1,2} = \pm\sqrt{C_1}, \omega_{3,4} = \pm\sqrt{C_2}, \omega_{5,6} = \pm\sqrt{C_3}. \tag{12.29}$$

If the controller parameter is set to zero $P = 0$ Nmrad$^{-1}$, $C_1$ is also zero. $C_2$ and $C_3$ are positive values. The first Eigenfrequency depends strongly on the value of $P$ and the initial position. With $P = 45$ Nmrad$^{-1}$ and $x_0 = (0, 0.1, 0.2, 0.3, 0.4, 0.5)$ m, the first Eigenfrequencies result to (81.7, 78.2, 75.4, 73.1, 69.8) Hz.

### *12.3.1 Identification of Nominal Belt Drive Parameters Using Cost Function Weights in Time Domain*

In real systems, it is often hard to get the correct controller values, therefore the controller value is identified. The axial stiffness $EA$ of the belt is also a parameter for the identification, since the real material behaviour usually differs from the values of data sheets due to tolerances. As measurement data, the angle $\varphi_1$ and torque $M_1$ of the driving gear from a reference simulation are used in order to test the identification algorithm. The reference angle $\varphi_{1d}$ was prescribed in form of small steps in order to get a good excitation of the undamped system i.e. shown in Fig. 12.4.

Due to high frequency in the angle signal of the driving gear, i.e. shown in Fig. 12.4, a sampling interval of 0.2 ms was chosen as simulation time.

In following, the influence of the number of surviving parameters is shown. The parameters are searched in the intervals $EA \in [1.5 \times 10^5, 2.5 \times 10^5]$ N and $P \in [30, 60]$ Nm rad$^{-1}$. The distribution of the cost function $e$ has several local minima, see

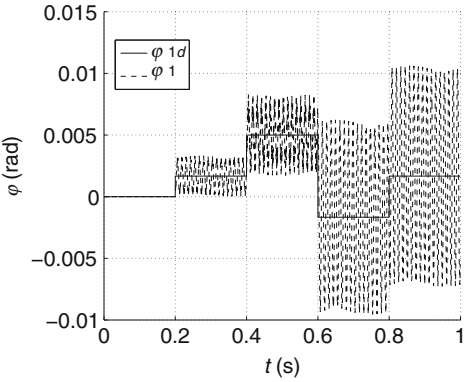



**Fig. 12.4** Step like reference angle $\varphi_{1d}$ and sinusoidal angle $\varphi_1$ of driving gear i.e. used for the computation of the cost function of the identification. The driving torque $M_1$ is proportional to the difference of these angles

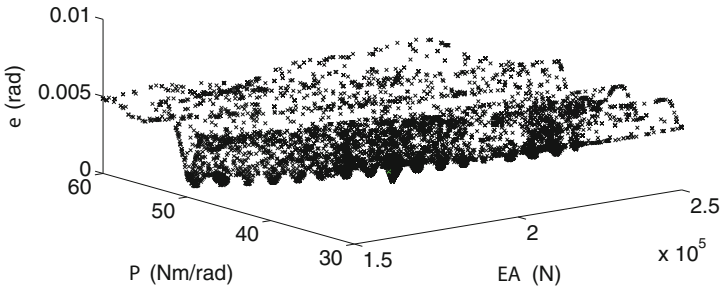



**Fig. 12.5** Distribution of the cost function $e$ computed in time domain, using (12.9), within limited parameter space $P$

Fig. 12.5. For the case of minimal distance factor is set to zero, the best parameter set was found correctly with eight surviving parameters after 1,700 computations. Tests with high minimal distance factor $d_{min,0} = 0.5$ and $|S^g| = 8$ are shown in [14]. With $|S^g| = 20$ and high minimal allowed distance factor $d_{min,0} = 0.5$, the identification of the parameters is shown in Fig. 12.6. Due to the high value, only 2,100 simulations are finally needed, even though $|S^g|$ is increased to maximal 20 survivors for each generation. In this case, the optimized parameters $EA_{opt} = 200,056$ N and $P_{opt} = 45.0145$ Nmrad$^{-1}$ are very close to the nominal parameters. The number of simulations is bigger in case of $d_{min,0} = 0.5$, so the value zero is better for this identification, see Table 12.2. The simulation error and the tested parameters during the identification are shown in Fig. 12.6. Remarkable are the high number of local minima in Fig. 12.6 with respect to the axial stiffness $EA$.

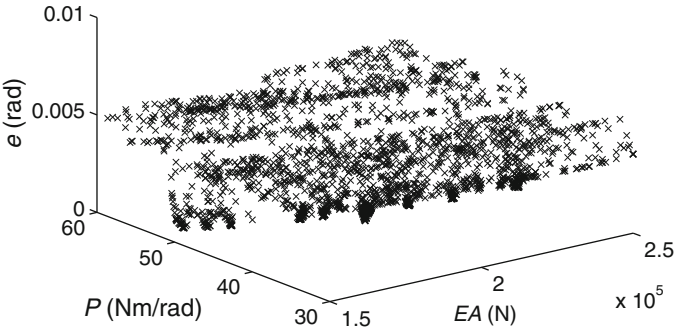



**Fig. 12.6** Cost function distribution obtained by (12.9) based on time domain from identification of axial stiffness $EA$ and controller gain $P$ using $d_{min,0} = 0.5$ and $|S^g| = 20$

**Table 12.2** Settings of identification algorithm using cost function in time domain

| Identification parameter | Value |
|---|---|
| Initial population size | 1,000 |
| Cardinality $\|S^g\|$ of surviving parameter set[a] | 8 and 20 |
| Number of children of each survivor | 8 |
| Number of generations $g$ | 12 |
| Range reduction factor $\xi$ | 0.5 |
| Minimal allowed distance[a] $d_{min,0}$ | 0 and 0.5 |

[a]Multiple identifications with these parameters are done for test purpose

The big influence of the axial stiffness to a phase shift of the gear angle is reason for this behavior. Nevertheless, with a sufficient number of initial values, survivors and child parameters, the original values can be retrieved easily.

### *12.3.2 Identification of Nominal Belt Drive Parameters Using Cost Function Weights in Frequency Domain*

In order to get faster convergence of the identification, the frequency dependent weight factors, described in (12.11) ff, are used. The modified distribution of the cost function in parameter space $P$ and an result[3] of the identification is computed. The optimal parameters $EA_{opt} = 200,416\text{N}$, $P_{opt} = 45.0077\,\text{Nm rad}^{-1}$, compare Figs. 12.7 and 12.8, located very close to the nominal parameters, are found in $3rd$ generation 2 after only 180 computations. The weights $w_k = 1$ is used in frequency intervals $I_1 = [70, 100]\text{Hz}$ and $I_2 = [450, 550]\text{Hz}$, wherein the Eigenfrequencies of the system are located. For all other frequency intervals, the weight factors $w_k = 0$ are applied. If the intervals are set extremely close to the Eigenfrequencies

[3] Inital population size $= 100$, $N_s = 10$, $N_c = 10$, $\xi = 0.5$, $d_{min,0} = 0.1$

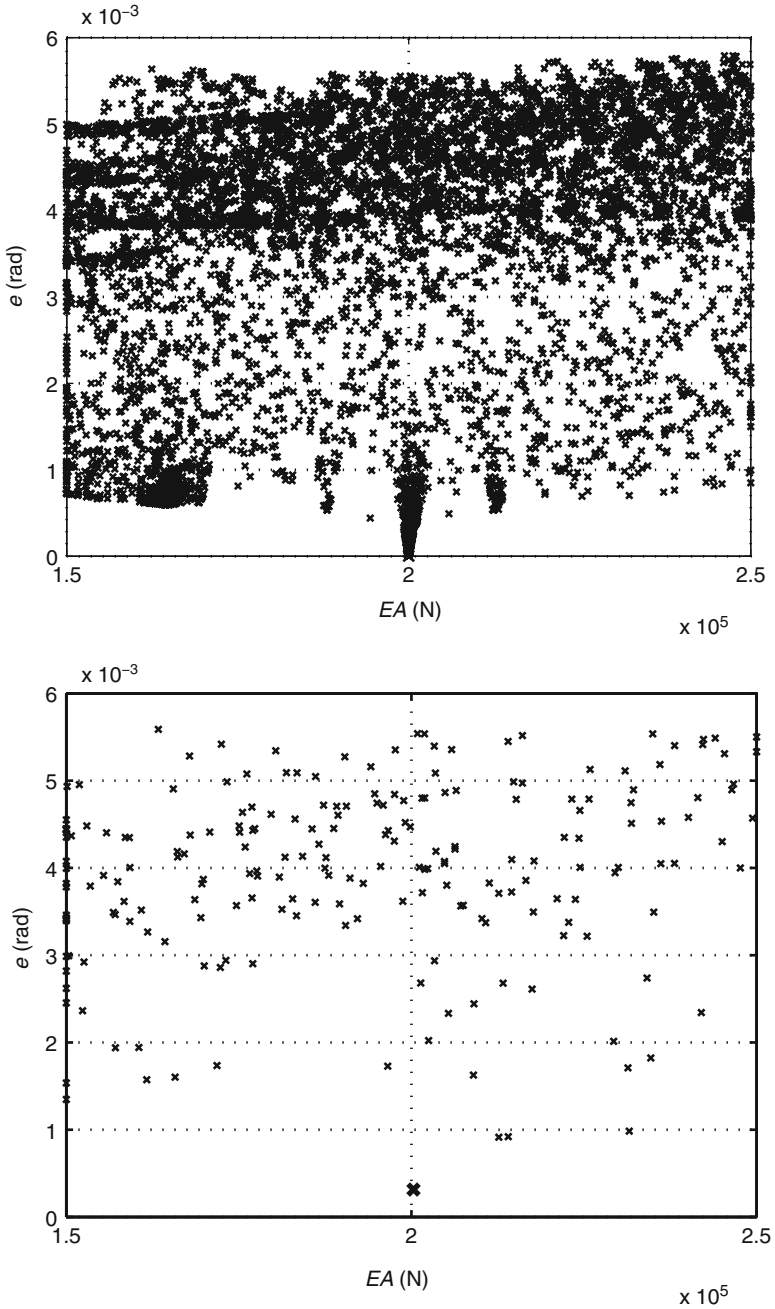



**Fig. 12.7** Cost function distribution (*top*) and result of the identification after only 180 simulations (*bottom*) obtained after use of frequency dependent weight factors $w_k$, *EA* nominal: 200, 000N, *EA* identified: 200, 416N

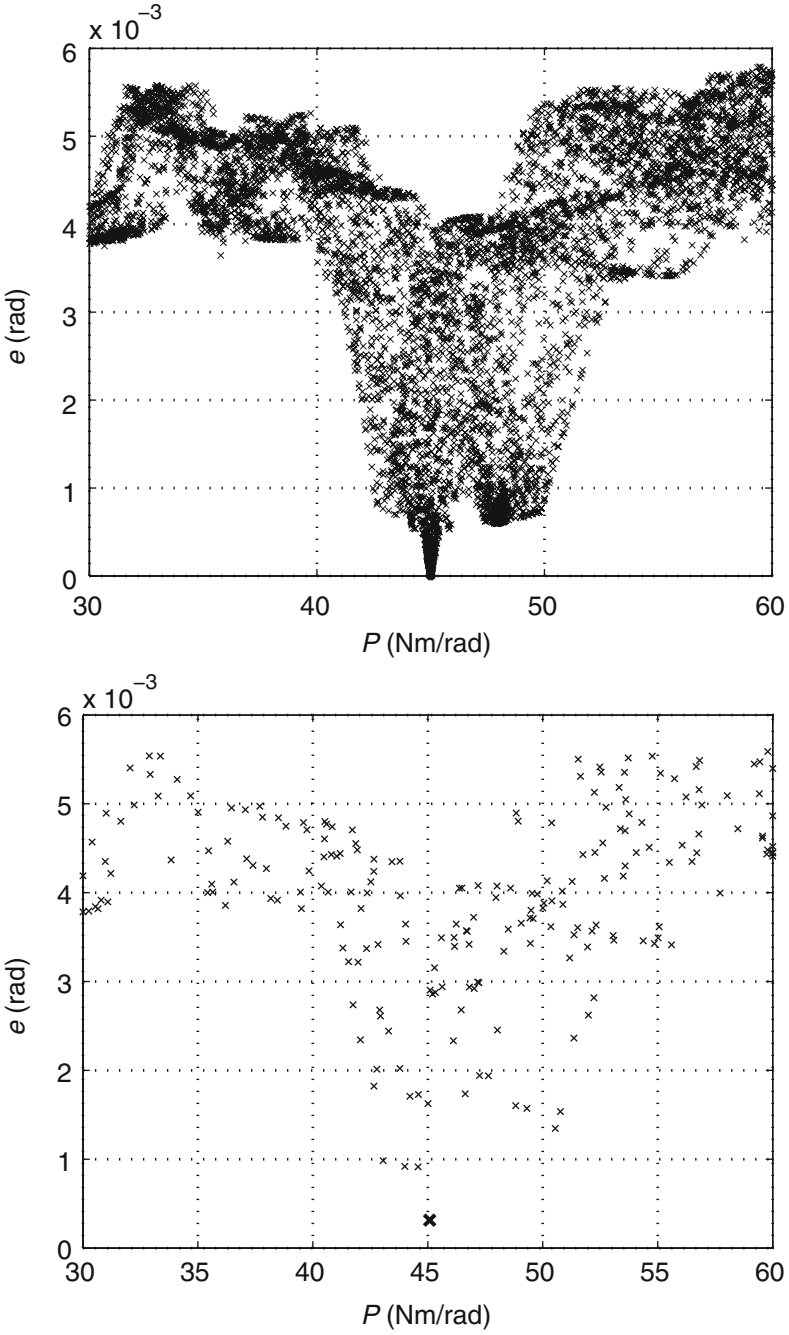



**Fig. 12.8** Cost function distribution (*top*) and result of the identification after only 180 simulations (*bottom*) obtained after use of frequency dependent weight factors $w_k$, $P$ nominal: 45 Nm rad$^{-1}$, $P$ identified: 45.0077 Nm rad$^{-1}$

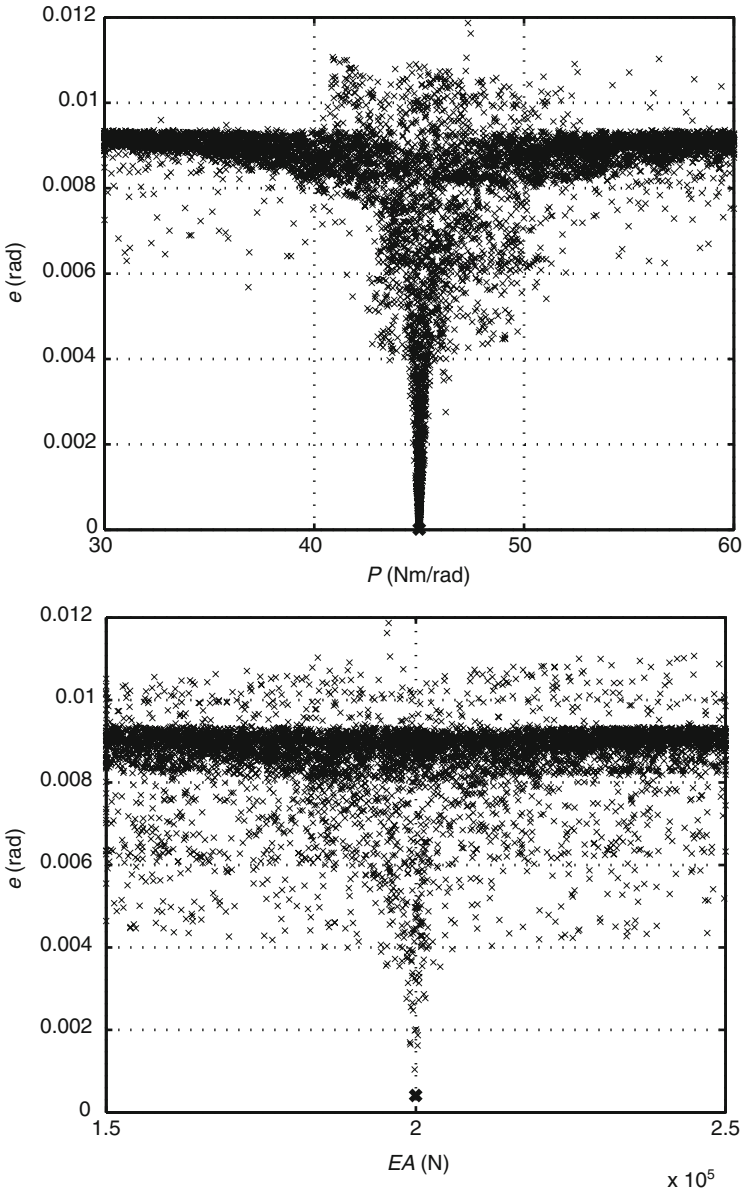



**Fig. 12.9** Dominant global minima in cost function distribution obtained after use of frequency dependent weight factors $w_k$, which are unequal zero only in interals very close to Eigenfrequencies, different weight factors are used ($I_1 = [80, 84]$Hz, $I_2 = [473, 476]$Hz, $w_k = 1 \forall k \in I_1$; $w_k = 3 \forall k \in I_2$; $w_k = 0 \forall k \notin I_1 \cup I_2$)

$I_1 = [80, 84]$Hz and $I_2 = [473, 476]$Hz, and the weights $w_k = 3$ for the interval $I_2$ the distribution of the cost function is more flat and the global minimum is more dominant, see Fig. 12.9.

## 12.4 Application of the Identification to a Real Robot with Parallel Kinematics

As final example, the algorithm is applied for an identification of parameters of a real robot, see Fig. 12.10, from recorded data containing noise and quantification effects. Nonlinear behavior of stiffnesses and friction, time delays, different robot kinematics and controller circuit are hard be investigated with conventional analytical identification methods.

The algorithm for identification identifies several parameters containing time delay within the environment of the multibody code HOTINT, which allows the simulation of complex robot motions in real time. As short summary of the simulation environment, see also [12, 13, 14], the robot simulation computes dynamic motions of serial robots with arbitrary number of rotational and translational robot links. Robots with parallel kinematics are also implemented in HOTINT.

In this section, the recorded motor torque contains high frequencies of oscillations, so this torque is chosen as output $\boldsymbol{y}_{mes}(t_j)$ according to (12.8). The simulated motor torque $\boldsymbol{y}_{sim}\left(\boldsymbol{\theta}^{gc},\ t_j\right)$ consists of the electric motor torque, viscous and static friction. The components of the parameter vectors and the limits of the definition of the parameter space $P \subset \mathbb{R}^5$ are shown in Table 12.3. In order to create difficult conditions for the identification, the search intervals of the parameters are chosen very large. Due to this reason, some of the child parameter vectors $\boldsymbol{\theta}^{gc}$ contain also overcritical values of the relative damping $\zeta_G > 1$. The time delay $\Delta T$ denotes the transfer times between measuring the motor torque and writing it into a data file. The electric motor torque is multiplied with the gear factor. This leads to an equivalent force which acts on the rail additionally to the gravitation force. A multiplication of the drive inertia with the square of the gear factor leads to equivalent mass $m_e$ of the moment of inertia of the motor results. The rail mass $m_{rail}$ is well known from the data sheet. The rail mass and the equivalent mass from drive inertia $J_D$ are connected by a spring damper element with stiffness $c_G$ and gear damping

$$d_G = 2\zeta_G \sqrt{\frac{m_e m_{rail} c_G}{m_e + m_{rail}}}, \tag{12.30}$$

representing the mechanical gear effects. For better physical interpretation, the relative gear damping $\zeta_G$ is identified. Constraint equations with Lagrange parameter are used to keep the rod lengths constant. The rods are mounted between rails and tool and are assumed as massless. The identification uses the settings shown in Table 12.4.

**Table 12.3** Limits of drive parameters of robot with parallel kinematics for identification

| Parameter | Symbol | $\theta_{i,min}$ | $\theta_{i,max}$ | Unit |
|---|---|---|---|---|
| Measurement delay | $\Delta T$ | 0 | 0.1 | s |
| Gear stiffness | $c_G$ | $1 \times 10^2$ | $3 \times 10^5$ | N/m |
| Rel. gear damping | $\zeta_G$ | 0 | 2 | 1 |
| Tool mass | $m_t$ | 1 | 2 | kg |
| Drive inertia | $J_D$ | $5 \times 10^{-5}$ | $15 \times 10^{-5}$ | kg m$^2$ |

**Table 12.4** Settings for identification of robot parameters

| Identification parameter | Value |
|---|---|
| Time window for Discrete Fourier Transformation | $[4, 7]$s |
| Frequency window $I$ | $[0, 50]$Hz |
| Nonzero frequency dependent weight factors $w_k \in I$ | 1 |
| Initial population size | 20 |
| Cardinality $\lvert S^g \rvert$ of surviving parameter set | 20 |
| Number of children of each survivor | 20 |
| Number of generations $g$ | 10 |
| Range reduction factor $\xi$ | 0.5 |
| Minimal allowed distance $d_{min,0}$ | 0 |

**Fig. 12.10** The parameters of a linear drive of a Festo tripod, see also http://www.festo.de, robot are identified. A highly dynamic pick and place motion was chosen to get a good excitation for the measurement

The cost function in frequency domain is used for the identification of several different parameters of the real robot. The Discrete Fourier Transformation of the cost function between 4 and 7 s is used to identify the drive parameters of the robot. The result of the identification is shown in Table 12.5. The identification progress and the simulation with the identified parameters from Table 12.5 and the measurement are shown in Figs. 12.11 and 12.12.

**Table 12.5** Identified parameters with cost function in frequency domain and frequency dependent weights

| Parameter | Symbol | Identification result | Unit |
|---|---|---|---|
| Measurement delay | $\Delta T$ | $2.489 \times 10^{-2}$ | s |
| Gear stiffness | $c_G$ | $2.252 \times 10^{5}$ | N/m |
| Rel. gear damping | $\zeta_G$ | $3.68 \times 10^{-1}$ | 1 |
| Tool mass | $m_t$ | 1.92 | kg |
| Drive inertia | $J_D$ | $8.6 \times 10^{-5}$ | kg m$^2$ |
| Cost function residual | $e^{gc}_{min}$ | 0.124 | Nm |
| Domain has lower residual | | | |

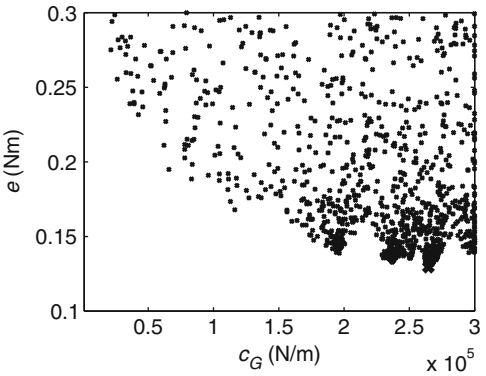



**Fig. 12.11** The gear stiffness $c_G$ is shown here with the corresponding cost function values $e$ using a very large search interval during the parameter identification of several parameters. The frequency dependent weights $w_k$ are set to 1 in the interval I = [0,50] Hz, and zero otherwise

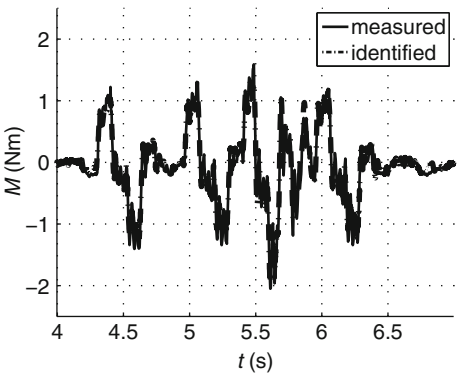



**Fig. 12.12** Simulated motor torque of simulations based on identified parameters, compare Table 12.3, and measured motor torque show high conformance

## 12.5 Conclusions

The chapter describes an algorithm for identification based on the theory of genetic optimization, which allows the identification of parameters of simple and complex mechatronic systems with large parameter ranges without algebraic manipulation of the dynamic equations. As extension of the computation of the cost function in the time domain, a strategy for searching optimal parameters in frequency domain is shown. The algorithm is tested by means of finding nominal parameters of a belt drive, which are exactly known. The use of frequency dependent weights improves the distribution of the cost function and the optimal parameters are found faster. The automatic parameter identification is used to search a list of user-defined parameters of a real robot. The result of the final simulation using the identified parameters is compared with recorded drive torque and shows high conformance.

**Acknowledgement** The authors appreciate the support of their work in the framework of the K2-Austrian Center of Competence in Mechatronics, ACCM. The authors like to acknowledge the cooperation on the present subject with the company KEBA AG.

## References

1. Arakawa A, Miyata KA (1996) Simultaneous optimization algorithm for determining both mechanical-system and controller parameters for positioning control mechanisms. In: Proceedings of the 4th international workshop on advanced motion control, Tsu, Mie, 1996
2. Bremer H, Pfeiffer F (1992) Elastische Mehrkörpersysteme. B G Teubner, Stuttgart
3. Cepon G, Manin L (2009) Introduction of damping into the flexible multibody belt-drive model: a numerical and experimental investigation. J Sound Vib 324:283–296
4. Dozier G, Homaifar A, Bikdash S (1998) Artificial potential field-based motion planning/navigation. Dynamic constrained optimization and simple genetic hill climbing. Simul 71:168–181
5. Fairweather AJ, Foster MP, Stone DA (2010) VRLA battery parameter, identification using Pseudo Random Binary Sequences (PBRS). In: Proceedings of the 5th IET international conference on power electronics, machines and drives, Brighton, 2010
6. Fitzpatrick JM, Grefenstette JJ (1988) Genetic algorithms in noisy environments. Mach Learn 3:101–120
7. Fliess M, Sira-Ramírez H (2003) An algebraic framework for linear identification. ESAIM: Control, Optim Calc Var 9:151–168
8. Fliess M, Join C, Sira-Ramírez H (2008) Non-linear estimation is easy. Int J Model Identif Control 4. doi:10.1504/IJMIC.2008.020996
9. Gerstmayr J (2009) HOTINT - A C++ environment for the simulation of multibody dynamics systems and finite elements. In: Proceedings of the eccomas thematic conference on multibody dynamics, Warsaw, pp 1–20
10. Hashim SZM, Tokhi MO, Darus IZM (2006) Active vibration control of flexible structures using genetic optimization. J Low Freq Noise Vib Act Control 25:195–207
11. Ljung L (1999) System identification theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River
12. Ludwig R, Gerstmayr J, Augdopler C, Mittermayer C (2008) Realistic robot simulation with flexible components. In: Proceedings of the 5th international conference on computational intelligence, robotics and autonomous systems (CIRAS), (2008), Linz, pp 187–192

13. Ludwig R, Gerstmayr J, Augdopler C, Mittermayer C (2008) Flexible robot with clearance. In: Proceedings of the 4th European conference on structural control, vol 2, Saint-Petersburg, pp 511–518
14. Ludwig R, Gerstmayr J (2011) Automatic parameter identification for generic robot models. In: Proceedings of the multibody dynamics eccomas thematics, Brussels, pp 1–16
15. Martínez JL, Mandow A, Morales J, Pedraza S, García-Cerezo A (2005) Approximating kinematics for tracked mobile robots. Int J Robotics Res 24:867–878
16. Pfeiffer F, Hölzl J (1995) Parameter identification for industrial robots. In: Proceedings of the IEEE international conference in robotics and automation, Nagoya, Japan, pp 1468–1476
17. De Queiroz MS (2009) An active identification method of rotor unbalance parameters. J Vib Control 15:1365–1374
18. Ramos F, Feliu V, Payo I (2008) Design of trajectories with physical constraints for very lightweight single link flexible Arms. J Vib Control 14:1091–1110
19. Shabana AA (1998) Dynamics of multibody systems, 2nd edn. Cambridge University Press, Cambridge, USA
20. Teng TK, Shieh JS, Chen CS (2003) Genetic algorithms applied in online autotuning PID parameters of a liquid – level control system. Trans Inst Meas Control 5:433–450
21. Xiaomin X, Qing S, Ling Z, Bin Z (2009) Parameter estimation and its sensitivity analysis of the MR damper hysteresis model using a modified genetic algorithm. J Intell Material Syst Struct 20:2089–2100
22. Zakharov A, Halasz S (2001) Parameter identification of a robot arm using genetic algorithms. Period Polytechn Ser El Eng 45:195–209
23. Ziegler F (1998) Technische Mechanik der festen und flüssigen Körper. Springer, Wien

# Chapter 13
# Crane Operators Training Based on the Real-Time Multibody Simulation

**Mohamad Ezral Baharudin, Pasi Korkealaakso, Asko Rouvinen, and Aki Mikkola**

**Abstract** This paper introduces a real-time multibody simulation approach. Two main sections have been described in depth and include a description of flexible bodies and modeling of a hydraulic system. In flexible bodies, the bodies are modelled using the floating frame of reference formulation. The equation of motion for the body is developed using the principle of virtual work. Penalty method is used when there are constraints in the mechanical system. The hydraulic system is modelled using lumped fluid theory. Two types of components, valves and hydraulic cylinders, are introduced for modelling. A numerical example is developed using two Craig-Bampton modes deformation modes modelled as flexible bodies.

## 13.1 Introduction

Products of mechanical industry, such as mobile machines and cranes, contain several different technology areas such as mechanics, actuators and control systems. These subsystems are in close interaction with each other as is shown in Fig. 13.1. In case of cranes, the actuators are often handled, in principle, as hydraulics. The hydraulic actuators are assembled on the mechanism to produce forces acting on the mechanism. The mechanism is typically a system of bodies, which converts the forces ofthe actuators into constrained motion. Electronics, together with the control algorithm that defines the way the structure behaves, can

M.E. Baharudin (✉) • A. Mikkola
Department of Mechanical Engineering, Lappeenranta University of Technology, Lappeenranta, Finland
e-mail: ezral.baharudin@lut.fi; aki.mikkola@lut.fi

P. Korkealaakso • A. Rouvinen
MeVEA Ltd, Laserkatu 6, Lappeenranta 53850, Finland
e-mail: pasi.korkealaakso@mevea.com; asko.rouvinen@mevea.com

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_13, 

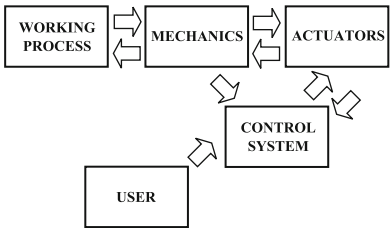



**Fig. 13.1** Real-time simulation model

be used in both open and closed loop control systems and are integrated into machine systems in order to increase productivity and ergonomics.

In most cases, solution time of the traditional simulation methods used in the product development processes are not synchronized to real-time. Accordingly, a simulation of a few seconds is allowed to take several hours of real time computation. In these systems, the control signals of the simulated system must be pre-defined and, for this reason, user interaction is described more or less experimentally based on measured data. When the simulation is executed, synchronous to real-time, the operator can produce a control signal during simulation. However, real-time solution requirements often force a simplification to be made in the simulation model. In practice, the real-time model can be considered as a trade-off between efficiency and accuracy.

The objective of this paper is to introduce a general simulation approach that can be applied for the real-time simulation of hydraulically driven cranes. The introduced approach is based on the use of the floating frame of reference formulation and is coupled with the lumped fluid theory, which allows for the description of hydraulic circuits. The floating frame of reference formulation can be used together with modal reduction methods. This feature allows for the optimization of the computational efficiency, such that solution time can be synchronized with real-time. In the section of numerical examples, the introduced simulation approach is applied to create real-time simulation models for two cranes.

## 13.2 Description of Flexible Bodies

In this section, the description of flexible bodies is shortly explained. In this study, the flexible bodies are modeled using the floating frame of reference formulation. The formulation can be applied to bodies that experience large rigid body translations and rotations; as well as elastic deformations. The method is based on describing deformations of a flexible body with respect to a frame of reference. The frame of reference, in turn, is employed to describe large translations and rotations. The deformations of a flexible body with respect to its frame of reference can be described with a number of methods, whereas in this study, deformation is described using linear deformation modes of the body. Deformation modes can be

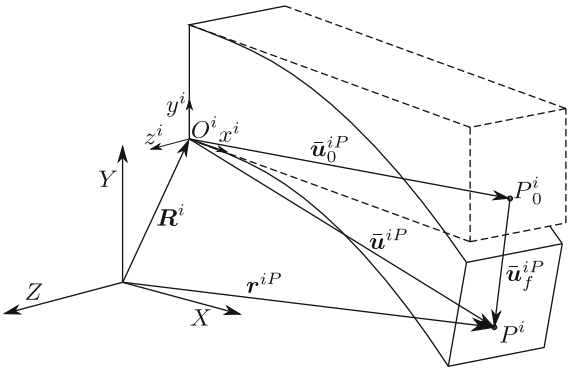



**Fig. 13.2** The position of the particle $P^i$ in global coordinate system

defined using a finite element model of the body [1, 2]. Figure 13.2 illustrates the position of particle $P^i$ in a deformed body $i$.

The position of particle $P^i$ of the flexible body $i$ can be described in a global coordinate system using the vector $\mathbf{r}^{P^i}$ as follows:

$$\mathbf{r}^{P^i} = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^{P^i} = \mathbf{R}^i + \mathbf{A}^i(\bar{\mathbf{u}}_0^{P^i} + \bar{\mathbf{u}}_f^{P^i}) \tag{13.1}$$

where $\mathbf{R}^i$ is the position vector of the frame of reference, $\mathbf{A}^i$ is the rotation matrix of body $i$, $\bar{\mathbf{u}}^{P^i}$ is the position vector of particle $P^i$ within the frame of reference, $\bar{\mathbf{u}}_0^{P^i}$ is the undeformed position vector of the particle within the frame of reference, and $\bar{\mathbf{u}}_f^{P^i}$ is the displacement of particle $P^i$ within the frame of reference due to the deformation of body $i$. In this study, the rotation matrix $\mathbf{A}^i$ is expressed using Euler parameters ${\theta^{E^i}}^{\mathrm{T}} = [\,\theta_0^{E^i} \quad \theta_1^{E^i} \quad \theta_2^{E^i} \quad \theta_3^{E^i}\,]^{\mathrm{T}}$ in order to avoid singular conditions which are problematic when three rotational parameters are used, such as in the case of Euler and/or Bryant angles [3]. The rotation matrix can be written using Euler parameters as follows:

$$\mathbf{A}^i = 2\begin{bmatrix} \frac{1}{2} - \left(\theta_2^{E^i}\right)^2 - \left(\theta_3^{E^i}\right)^2 & \theta_1^{E^i}\theta_2^{E^i} - \theta_0^{E^i}\theta_3^{E^i} & \theta_1^{E^i}\theta_3^{E^i} + \theta_0^{E^i}\theta_2^{E^i} \\ \theta_1^{E^i}\theta_2^{E^i} + \theta_0^{E^i}\theta_3^{E^i} & \frac{1}{2} - \left(\theta_1^{E^i}\right)^2 - \left(\theta_3^{E^i}\right)^2 & \theta_2^{E^i}\theta_3^{E^i} - \theta_0^{E^i}\theta_1^{E^i} \\ \theta_1^{E^i}\theta_3^{E^i} - \theta_0^{E^i}\theta_2^{E^i} & \theta_2^{E^i}\theta_3^{E^i} + \theta_0^{E^i}\theta_1^{E^i} & \frac{1}{2} - \left(\theta_1^{E^i}\right)^2 - \left(\theta_2^{E^i}\right)^2 \end{bmatrix} \tag{13.2}$$

The following mathematical constraint must be taken into consideration when Euler parameters are applied:

$$\left(\theta_0^{E^i}\right)^2 + \left(\theta_1^{E^i}\right)^2 + \left(\theta_2^{E^i}\right)^2 + \left(\theta_3^{E^i}\right)^2 = 1 \tag{13.3}$$

The deformation vector $\bar{\mathbf{u}}_f^{P^i}$ can be described using a linear combination of the deformation modes

$$\bar{\mathbf{u}}_f^{P^i} = \Psi_R^{P^i} \mathbf{q}_f^i \tag{13.4}$$

Where $\Psi_R^{P^i}$ is the modal matrix whose columns describe the translation of particle $P^i$ within the assumed deformation modes of the flexible body $i$ [4], and $\mathbf{q}_f^i$ is the vector of elastic coordinates. Consequently, the generalized coordinates that uniquely define the position of point $P^i$ can be represented with vector $\mathbf{p}^i$ as follows:

$$\mathbf{p}^{i\mathrm{T}} = \begin{bmatrix} \mathbf{R}^{i\mathrm{T}} & \theta^{E^i\mathrm{T}} & \mathbf{q}_f^{i\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{13.5}$$

The velocity of particle $P^i$ can be obtained by differentiating the position description (13.1) with respect to time

$$\dot{\mathbf{r}}^{P^i} = \dot{\mathbf{R}}^i - \mathbf{A}^i \left( \tilde{\bar{\mathbf{u}}}_0^{P^i} + \tilde{\Psi}_R^{P^i} \mathbf{q}_f^i \right) \bar{\omega}^i + \mathbf{A}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \tag{13.6}$$

Where $\bar{\omega}^i$ is the vector of local angular velocities. In (13.6), the generalized velocity vector can be defined

$$\dot{\mathbf{q}}^{i\mathrm{T}} = \begin{bmatrix} \dot{\mathbf{R}}^{i\mathrm{T}} & \bar{\omega}^{i\mathrm{T}} & \dot{\mathbf{q}}_f^{i\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{13.7}$$

By differentiating (13.6) with respect to time, the following formulation for the acceleration of particle $P^i$ can be obtained:

$$\ddot{\mathbf{r}}^{P^i} = \ddot{\mathbf{R}}^i + \mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + \mathbf{A}^i \tilde{\dot{\bar{\omega}}}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\bar{\omega}}^i \dot{\bar{\mathbf{u}}}^{P^i} + \mathbf{A}^i \ddot{\bar{\mathbf{u}}}^{P^i} \tag{13.8}$$

where $\tilde{\bar{\omega}}^i$ is a skew-symmetric representation of the angular velocity of the body in the frame of reference, $\ddot{\mathbf{R}}^i$ is the vector that defines the translational acceleration of the frame of reference, $\mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i}$ is the normal component of acceleration, $\mathbf{A}^i \tilde{\dot{\bar{\omega}}}^i \bar{\mathbf{u}}^{P^i}$ is the tangential component of acceleration, $2\mathbf{A}^i \tilde{\bar{\omega}}^i \dot{\bar{\mathbf{u}}}^{P^i}$ is the Coriolis component of acceleration and $\mathbf{A}^i \ddot{\bar{\mathbf{u}}}^{P^i}$ is the acceleration of particle $P^i$ due to the deformation of body $i$.

When deformation modes are used with the floating frame of reference, rotations due to body deformation are usually ignored. However, in order to compose all of the basic constraints, rotation due to body deformation must be accounted for. The vector $\bar{\mathbf{v}}_f^i$ due to deformation at the location of particle $P^i$ within the frame of reference can be expressed

$$\bar{\mathbf{v}}_f^i = \mathbf{A}_f^{P^i} \bar{\mathbf{v}}^i \tag{13.9}$$

Where $\bar{\mathbf{v}}^i$ is defined in the undeformed state at the location of particle $P^i$, and $\mathbf{A}_f^{P^i}$ is a rotation matrix that describes the orientation due to deformation at the location of particle $P^i$ with respect to the reference frame. Note that all components in (13.9) are expressed in the reference frame. The rotation matrix $\mathbf{A}_f^{P^i}$ can be expressed as follows:

$$\mathbf{A}_f^{P^i} = \mathbf{I} + \tilde{\varepsilon}^{P^i} \tag{13.10}$$

In (13.10), $\mathbf{I}$ is a $(3 \times 3)$ identity matrix and $\tilde{\varepsilon}^{P^i}$ is a skew symmetric form of the rotation change caused by deformation. Rotation changes due to deformation can be represented as the following:

$$\varepsilon^{P^i} = \Psi_\theta^{P^i} \mathbf{q}_f^i \tag{13.11}$$

Where $\Psi_\theta^{P^i}$ is the modal transformation matrix whose columns describe rotation coordinates of point $P^i$ within the assumed deformation modes of the flexible body $i$ [4].

### 13.2.1 Equations of Motion

The equations of motion can be developed using the principle of virtual work. When the floating frame of reference formulation is employed the virtual work done by inertial forces can be written as follows:

$$\delta W^{i^i} = \int_{V^i} \rho^i \delta \mathbf{r}^{P^i{}^{\mathrm{T}}} \ddot{\mathbf{r}}^{P^i} \, dV^i \tag{13.12}$$

where $\delta \mathbf{r}^{P^i}$ is the virtual displacement of the position vector of a particle, $\ddot{\mathbf{r}}^{P^i}$ is the acceleration vector of the particle defined in (13.8), $\rho^i$ is density of body $i$, and $V^i$ is volume of body $i$. The virtual displacement of the position vector can be expressed in terms of virtual displacement of generalized coordinates as follows:

$$\delta \mathbf{r}^{P^i{}^{\mathrm{T}}} = \begin{bmatrix} \delta \mathbf{R}^{i^{\mathrm{T}}} & \delta \theta^{i^{\mathrm{T}}} & \delta \mathbf{q}_f^{i^{\mathrm{T}}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \\ -\tilde{\mathbf{u}}^{P^i{}^{\mathrm{T}}} & \mathbf{A}^{i^{\mathrm{T}}} \\ \Psi_R^{P^i{}^{\mathrm{T}}} & \mathbf{A}^{i^{\mathrm{T}}} \end{bmatrix} \tag{13.13}$$

By substituting the virtual displacement of position vector (13.13) into the equation of virtual work of the inertial forces (13.12), and by separating the terms related to acceleration from the terms related quadratically to velocities, the following equation for the virtual work of inertial forces can be obtained:

$$\delta W^{i^i} = \delta \mathbf{q}^i \left[ \mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{v^i} \right] \tag{13.14}$$

Where $\mathbf{M}^i$ is the mass matrix and $\mathbf{Q}^{v^i}$ is the quadratic velocity vector. The mass matrix can be expressed as follows:

$$\mathbf{M}^i = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A}^i \tilde{\bar{\mathbf{u}}}^{P^i} & \mathbf{A}^i \Psi_R^{P^i} \\ & \tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\mathbf{u}}}^{P^i} & -\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \Psi_R^{P^i} \\ sym & & \Psi_R^{P^i \mathrm{T}} \Psi_R^{P^i} \end{bmatrix} dV^i \tag{13.15}$$

And, correspondingly, the quadratic velocity vector takes the form

$$\mathbf{Q}^{v^i} = \int_{V^i} \rho^i \begin{bmatrix} \mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \\ -\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} - 2\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \\ \Psi_R^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\Psi_R^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \end{bmatrix} dV^i \tag{13.16}$$

The virtual work of the externally applied forces can be written as:

$$\delta W^{e^i} = \int_{V^i} \delta \mathbf{r}^{P^i \mathrm{T}} \mathbf{F}^{P^i} dV^i = \delta \mathbf{q}^{i \mathrm{T}} \mathbf{Q}^{e^i} \tag{13.17}$$

Where $\mathbf{F}^{P^i}$ is external force per unit mass and $\mathbf{Q}^{e^i}$ is the vector of generalized forces which can be expressed as follows:

$$\mathbf{Q}^{e^i} = \begin{bmatrix} \sum_{j=1}^{n_F} \mathbf{F}_j^i \\ \sum_{j=1}^{n_F} \tilde{\bar{\mathbf{u}}}_j^i \mathbf{A}^{i \mathrm{T}} \mathbf{F}_j^i \\ \sum_{j=1}^{n_F} \Psi_j^{i \mathrm{T}} \mathbf{A}^{i \mathrm{T}} \mathbf{F}_j^i \end{bmatrix} \tag{13.18}$$

where $\mathbf{F}_j^i$ is the j-th force component acting on body $i$, $\tilde{\bar{\mathbf{u}}}_j^i$ is a skew symmetric matrix of the location vector of the j-th force components, and $\Psi_j^i$ includes the terms of the modal matrix associated with the node to which the j-th force component applies.

The elastic forces can be defined using the modal stiffness matrix $\mathbf{K}^i$ and modal coordinates. The modal stiffness matrix is associated with the modal coordinates and the matrix can be obtained from the conventional finite element approach using the component mode synthesis technique [4]. The virtual work of elastic forces can be written as follows:

$$\delta W^{s^i} = \delta \, \mathbf{q}_f^{i \mathrm{T}} \mathbf{K}^i \mathbf{q}_f^i \tag{13.19}$$

Accordingly, the vector of elastic forces can be represented as follows:

$$\mathbf{Q}^{f^i} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix} \tag{13.20}$$

Using (13.14), (13.17) and (13.19), the equation of virtual work, including inertial, external and internal force components, can be written as follows:

$$\delta \mathbf{q}^i \left[ \mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{v^i} + \mathbf{Q}^{f^i} - \mathbf{Q}^{e^i} \right] = 0 \tag{13.21}$$

The terms inside the brackets can be used to form unconstrained Newton-Euler equations as follows:

$$\begin{aligned}
&\begin{bmatrix} \int_{V^i} \rho^i \mathbf{I} dV^i & -\int_{V^i} \rho^i \mathbf{A}^i \tilde{\bar{\mathbf{u}}}^{P^i} dV^i & \int_{V^i} \rho^i \mathbf{A}^i \boldsymbol{\Psi}_R^{P^i} dV^i \\ & \int_{V^i} \rho^i \tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\mathbf{u}}}^{P^i} dV^i & -\int_{V^i} \rho^i \tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \boldsymbol{\Psi}_R^{P^i} dV^i \\ sym & & \int_{V^i} \rho^i \boldsymbol{\Psi}_R^{P^i \mathrm{T}} \boldsymbol{\Psi}_R^{P^i} dV^i \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}}^i \\ \dot{\bar{\omega}}^i \\ \ddot{\mathbf{q}}_f^i \end{bmatrix} \\
&= \begin{bmatrix} \int_{V^i} \mathbf{F}^{P^i} dV^i \\ \int_{V^i} \tilde{\bar{\mathbf{u}}}^{P^i} \mathbf{A}^{i\mathrm{T}} \mathbf{F}^{P^i} dV^i \\ \int_{V^i} \boldsymbol{\Psi}_R^{P^i \mathrm{T}} \mathbf{A}^{i\mathrm{T}} \mathbf{F}^{P^i} dV^i \end{bmatrix} \\
&- \begin{bmatrix} \int_{V^i} \rho^i \left( \mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \\ \int_{V^i} \rho^i \left( -\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} - 2\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \\ \int_{V^i} \rho^i \left( \boldsymbol{\Psi}_R^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\boldsymbol{\Psi}_R^{P^i \mathrm{T}} \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \end{bmatrix} \\
&- \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix}
\end{aligned} \tag{13.22}$$

Equations of motion in this form are referred to as Generalized Newton-Euler equations in [4], where Newton-Euler equations of rigid bodies are extended to flexible bodies.

### 13.2.2 Integration of the Equations of Motion

Due to the use of Generalized Newton-Euler equations as a description of dynamics, the equations of motion are expressed using the angular velocity and angular acceleration vectors. Equation (13.22) can be solved to obtain angular accelerations in the body frame, which can be integrated with angular velocities. However, the problem arises when the coordinates describing the orientation of the body have to be solved. This is due to the fact that angular velocities cannot be directly integrated with the parameters which uniquely describe the orientation of the body. For this reason, a new set of variables $\mathbf{p}$ is defined, containing the orientation coordinates of the body reference frame. In order to integrate the position level coordinates, a relationship between the first time derivative of Euler parameters and the vector of angular velocities defined in the body reference frame can be made through the following linear expression:

$$\dot{\theta}^{E^i} = \frac{1}{2}\bar{\mathbf{G}}^{i\mathrm{T}}\bar{\omega}^i \tag{13.23}$$

where the velocity transformation matrix $\bar{\mathbf{G}}^i$ can be written as follows:

$$\bar{\mathbf{G}}^i = \begin{bmatrix} -\theta_1^{E^i} & \theta_0^{E^i} & \theta_3^{E^i} & -\theta_2^{E^i} \\ -\theta_2^{E^i} & -\theta_3^{E^i} & \theta_0^{E^i} & \theta_1^{E^i} \\ -\theta_3^{E^i} & \theta_2^{E^i} & -\theta_1^{E^i} & \theta_0^{E^i} \end{bmatrix} \tag{13.24}$$

The time derivatives of the body variables to be integrated can be stated using vector $\dot{\mathbf{p}}$ as follows:

$$\dot{\mathbf{p}}^{i\mathrm{T}} = \left[ \dot{\mathbf{R}}^{i\mathrm{T}} \dot{\theta}^{E^i\mathrm{T}} \dot{\mathbf{q}}_f^{i\mathrm{T}} \right]^{\mathrm{T}} \tag{13.25}$$

which can be integrated to obtain position level generalized coordinates $\mathbf{p}$.

### 13.2.3 Description of Constrained Mechanical Systems

In this section, the penalty method used in this study is briefly described. Mechanical joints that restrict motion possibilities of interconnected bodies can be described with constraint equations [5]. To satisfy a set of $m$ constraint equations related to generalized coordinates, the following equation must be fulfilled:

$$\mathbf{C}(\mathbf{q}, t) = 0 \tag{13.26}$$

where $\mathbf{C}$ is a vector of constraints of the system and $t$ is time. In the penalty method, the constraints are combined to the equations of motion by employing penalty terms. This procedure leads to a set of $n$ differential equations as follows:

$$\left(\mathbf{M}+\mathbf{C}_q^{\mathrm{T}}\alpha\mathbf{C}_q\right)\ddot{\mathbf{q}}=\mathbf{Q}^e-\mathbf{Q}^v-\mathbf{Q}^f-\mathbf{C}_q^{\mathrm{T}}\alpha\left(\mathbf{Q}^c+2\Omega\mu\dot{\mathbf{C}}+\Omega^2\mathbf{C}\right) \tag{13.27}$$

where $\mathbf{C}_q$ is the Jacobian matrix of the constraint equations and $\alpha$, $\Omega$ and $\mu$ are $m \times m$ diagonal matrices and contain penalty terms, natural frequencies and damping ratios for constraints, respectively. If the penalty terms are equivalent to each constraint, the matrices are identity matrices multiplied with a constant penalty factor [6].

A drawback associated with the penalty method is that large numerical values for penalty factors must be used, which may lead to numerical ill-conditioning and round-off errors. However, the method can be improved by adding penalty terms or correction terms, which are zero when constraint equations are fulfilled. Using this approach, equations of motion can be written as follows:

$$\left(\mathbf{M}+\mathbf{C}_q^{\mathrm{T}}\alpha\mathbf{C}_q\right)\ddot{\mathbf{q}}=\mathbf{Q}^e-\mathbf{Q}^v-\mathbf{Q}^f-\mathbf{C}_q^{\mathrm{T}}\alpha\left(\mathbf{Q}^c+2\Omega\mu\dot{\mathbf{C}}+\Omega^2\mathbf{C}\right)+\mathbf{C}_q^{\mathrm{T}}\lambda^* \tag{13.28}$$

where $\lambda^*$ is the vector of penalty forces. Since the exact values of $\lambda^*$ are not known in advance, an iterative procedure should be used as follows:

$$\lambda_{i+1}^*=\lambda_i^*-\alpha\left(\mathbf{C}_q\ddot{\mathbf{q}}_i+\mathbf{Q}^c+2\Omega\mu\dot{\mathbf{C}}+\Omega^2\mathbf{C}\right) \tag{13.29}$$

where $\lambda^* = 0$ is used for the first iteration. Using this equation, the forces caused by errors in constraint equations at iteration $i + 1$ can be defined and compensated. In this case, the penalty terms do not need to have large numerical values. An iterative procedure can be applied directly to (13.28), which leads to the following expression:

$$\left(\mathbf{M}+\mathbf{C}_q^{\mathrm{T}}\alpha\mathbf{C}_q\right)\ddot{\mathbf{q}}_{i+1}=\mathbf{M}\ddot{\mathbf{q}}_i-\mathbf{C}_q^{\mathrm{T}}\alpha\left(\mathbf{Q}^c+2\Omega\mu\dot{\mathbf{C}}+\Omega^2\mathbf{C}\right) \tag{13.30}$$

In the case of the first iteration, $\mathbf{M}\ddot{\mathbf{q}}_0=\mathbf{Q}^e-\mathbf{Q}^v-\mathbf{Q}^f$, the leading matrix of (13.29) is a symmetric and positive definite, which makes the solution of the equation efficient. This formulation behaves satisfactorily also in singular configurations of a mechanical system.

## 13.3 Modelling of Actuators

In this study, crane systems are assumed to be driven with hydraulic actuators. Hydraulic actuators can be modelled using the lumped fluid theory, in which the hydraulic circuit is divided into volumes where pressures are assumed to be equally

distributed. In this approach, the pressure waves in pipes and hoses are assumed to be insignificant [7]. The hydraulic pressure in each hydraulic volume $i$ can be described as

$$\dot{p}_l = \frac{B_{ei}}{V_i} \sum_{j=1}^{n_c} Q_{ij} \tag{13.31}$$

where $B_{ei}$ is the effective bulk modulus of volume $i$, (13.31), $Q_{ij}$ is the outgoing or incoming flow rate of volume $i$ and $n_c$ is the total number of flows related to volume $i$. The effective bulk modulus defines the flexibility of the hydraulics and it can be calculated as follows:

$$B_{ei} = \frac{1}{\frac{1}{B_{oil}} + \sum_{j=1}^{n_c} \frac{V_j}{V_i B_j} + \sum_{k=1}^{n_h} Q_{ij} \frac{V_k}{V_i B_k}} \tag{13.32}$$

In (13.32), $n_h$ is the total number of pipes and hoses related to volume $i$. The bulk modulus of oil, $B_{oil}$, accounts for the amount of non-dissolving air in oil and it is a function of pressure. The maximum value is typically $B_{oilmax} = 1.6e9Pa$. It is important to note that the bulk modulus $B_j$ of component $j$ is also dependent on the component type.

### 13.3.1 Modelling of Valves

For modelling purposes, a valve is assumed to consist of several adjustable restrictor valves which can each be modelled separately [8]. With small pressure differences (pressure difference $< 1$ bar), the flow over the restrictor is thought to be laminar, whereas with larger differences, it is thought to be turbulent. When using the semi-empiric modelling method, the flow over the restrictor can be written as follows:

$$Q = C_v U \sqrt{dp} \tag{13.33}$$

where $C_v$ is the semi-empiric flow rate constant and defines the size of the valve and $U$ is a variable that defines the spool or poppet position. For a number of valve types, the variable $U$ can be defined using a first order differential equationas follows:

$$\dot{U} = \frac{U_{ref} - U}{\tau} \tag{13.34}$$

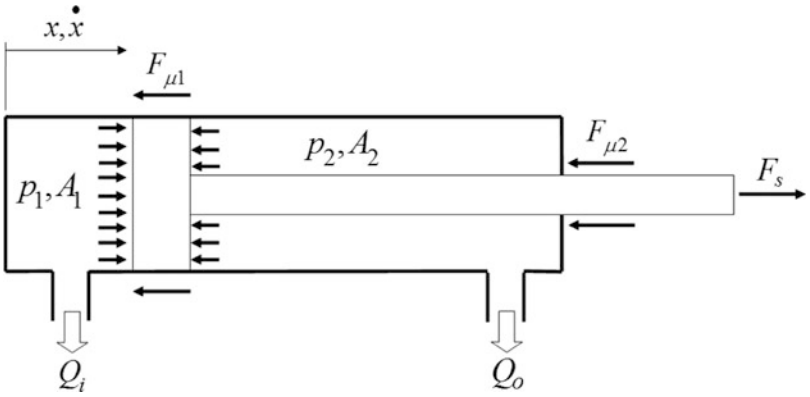



**Fig. 13.3** Hydraulic cylinder transform hydraulic pressure into a mechanical force

where $U_{ref}$ is the spool reference position and $t$ time constant describing the dynamics of the valve spool.

### 13.3.2 Modelling of Hydraulic Cylinders

A hydraulic cylinder can be modeled by simply using the dimensions of the cylinder and the pressure obtained from (13.31). The motion of the hydraulic cylinder produces a flow rate to the hydraulic volume $I$ as follows

$$\begin{aligned} Q_{jA} &= -\dot{x}A_A \\ Q_{jB} &= \dot{x}A_B \end{aligned} \tag{13.35}$$

where $A_A$ is the area of cylinder piston side and $A_B$ is the area on cylinder piston rod side, $\dot{x}$ is the velocity of the stroke of the cylinder, as depicted in Fig. 13.3.

The force produced by the hydraulic cylinder can be written as

$$F_s = p_1A_1 - p_2A_2 - F_\mu \tag{13.36}$$

where $F_\mu$ is the total friction force of the cylinder and $p_1$ and $p_2$ are pressures acting in the cylinder chambers. The friction force is a function of pressures, cylinder efficiency, $\eta$ and velocity. The friction force can be described in a simple case as follows:

$$F_\mu = (p_1A_1 - p_2A_2)(1 - \eta)f(\dot{x}) \tag{13.37}$$

The velocity dependent co-efficient, $f(\dot{x})$, can be described using a spline-curve as shown in Fig. 13.4.

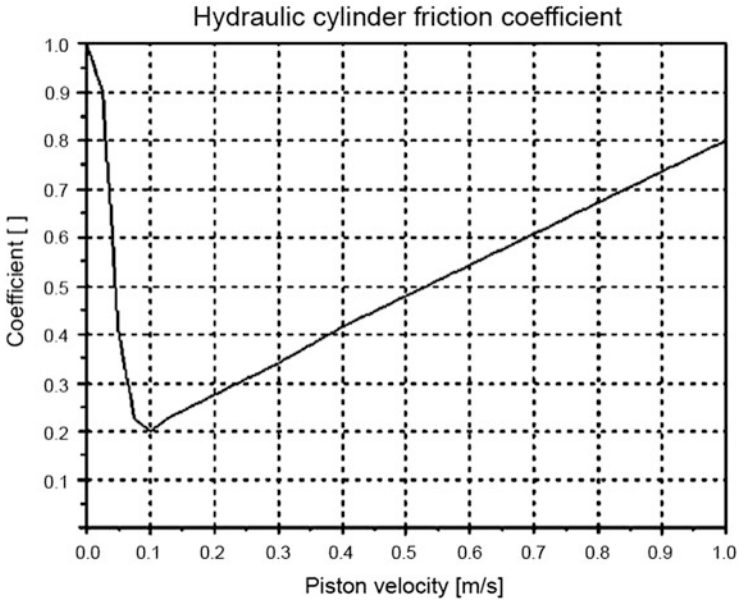



**Fig. 13.4** The velocity dependent friction co-efficient

### *13.3.3 Numerical Example*

The numerical example is based on a Liebherr LTM 1030 mobile crane and is shown in Fig. 13.5. The simulation model of the crane consists of eight bodies, of which four are modelled as flexible bodies using two Craig-Bampton deformation modes for each body. The model includes several force components related to wind loads of hooks and booms, hydraulics, the engine and a rope system using a particle based approach is modelled as well.

Since the model under investigation is used in training simulators, the hydraulic model is a simplified version of the actual hydraulics shown in Fig. 13.6.

Hydraulic circuits consist of hydraulic cylinders, pressure compensated proportional directional valves, pressure relief valves and pumps – in case of lift circuits, two counter balance valves are used. Figures 13.7 and 13.8 show examples of a simple work cycle, in Fig. 13.7 the valve control signal and spool opening are presented. Figure 13.8 presents the flow rates through the lift circuit valve and the pressure rates of cylinder chambers are shown in Fig. 13.9.

The simulation of the system was modelled using MeVEA Real-Time Simulation environment. The environment offers the possibility of off-line simulation for more detailed models or, alternatively, real-time simulation and visualization for simplified models [9]. The environment is compatible with MeVEA Full Mission Solution, which offers a motion platform and visualization environment combined with user interface – including an operators seat, joysticks, and pedal etc. case specific instrumentation [10].

**Fig. 13.5** Mobile crane in virtual construction site

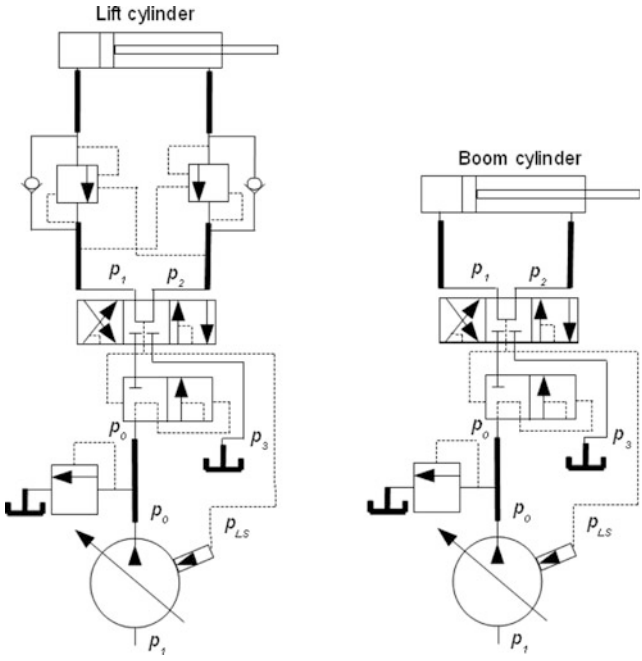



**Fig. 13.6** Simplified hydraulic system of a mobile crane

## 13.4 Simulation Environment

In order to simulate the actual environment of the mobile crane, the dynamic model must be presented with visualization and physical feedback. This implementation required a number of mechanical actuators, interfaces, software and a high performance computer as shown in Fig. 13.10.

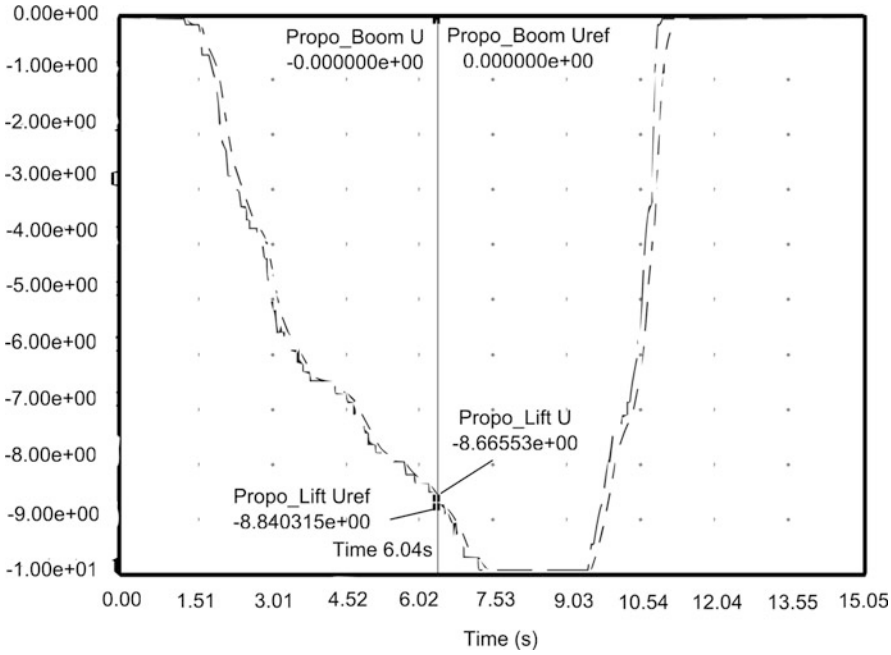



**Fig. 13.7** Lift circuit valve control signal and reference control signal

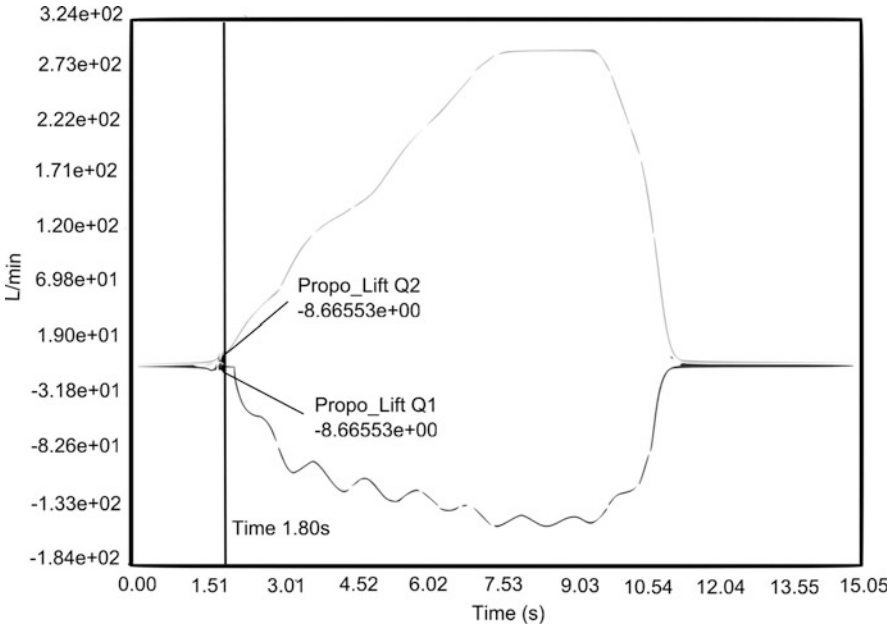



**Fig. 13.8** Flow rates through the lift valve during work cycle

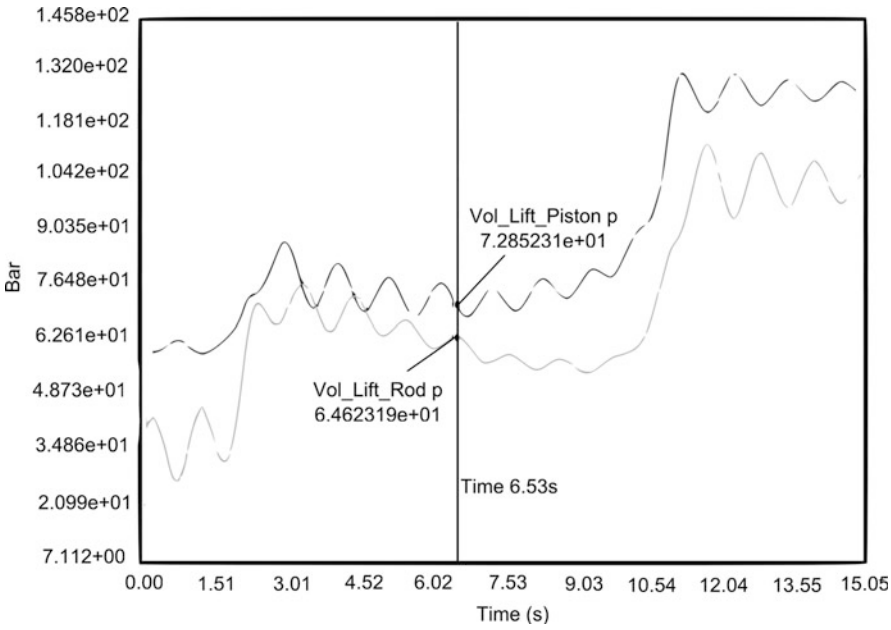



**Fig. 13.9** Lift circuit cylinder pressure rates

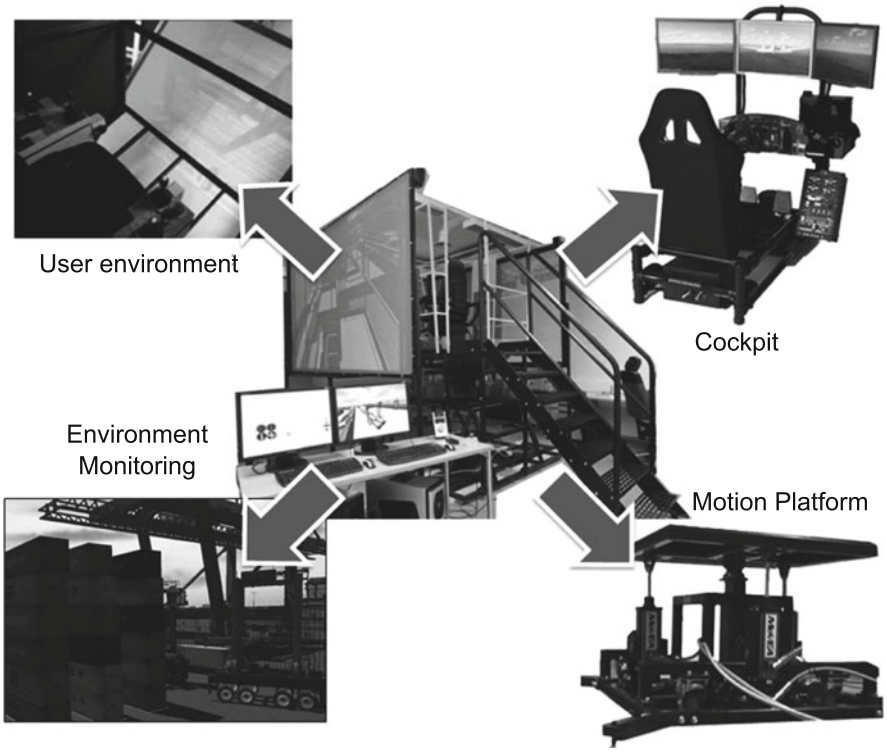



**Fig. 13.10** Real-time simulation environment

The high performance computer is required to solve equations of motion and hydraulic circuits in real-time. The simulation results can be sent out to the actuators of a motion platform via I/O interfaces and a computer. These are the most critical aspects, where the response time should be minimized in order to sustain the feel and sensation of real-time feedback. The accuracy of feedback and parameters involved in the numerical analysis should also be taken into account.

The visualization of the working environment was carried out using 3D development software. All the machine components were designed with the actual dimensions, and to get a realistic result, the actual environment images were also added. All the components were treated separately in such that they have their own dedicated local coordinate system. These local coordinate systems were synchronised with the dynamic model system in a computer and were connected to another computer, which acted as a main controller for receiving and controlling all of the input and output data. The main controller reacted based on the calculation obtained from the dynamic model. It controls the position and velocity of the actuators, as well as visualization aspects.

The simulator motion platform used in this project has six DOFs. This platform has translational and rotational movements. Additional actuator may require a high performance computer in order to ensure that the modelling process is sufficient enough to generate fast data. Accurate inverse kinematic models of the platform also need to be developed to get correct acceleration rates.

## 13.5 Conclusion

In the real-time simulation, a machine needs to be considered as a coupled system that consists of mechanical components and actuators. This study introduced a general simulation approach that can be applied for the real-time simulation of hydraulically driven cranes. The introduced approach was based on the use of the floating frame of reference formulation and was coupled with the lumped fluid theory for the modeling of hydraulic circuits. The floating frame of reference formulation can be used together with modal reduction methods. The introduced simulation approach was applied to create real-time simulation models of a mobile crane. The simulation model of the crane consists of eight bodies, of which four are modelled as flexible bodies using two Craig-Bampton deformation modes for each body. The model includes several force components related to the wind load of hooks and booms, hydraulics, the engine and modelling of rope system using a particle based approach. The simulation model of the crane was embedded into real-time simulation environment that consists of visualization, motion platform and an I/O interface. The real-time simulation environment allows for merging a user to the simulation model in a realistic manner.

## References

1. Haug EJ, Yoo WS (1986) Dynamics of flexible mechanical systems using vibration and static correction modes. J Mech Transm Automat Design 108:315–322
2. Haug EJ, Yoo WS (1986) Dynamics of articulated structures. Part I. Theory. J Struct Mech 14(1):105–126
3. Chung IS, Nikravesh PE (1982) Application of Euler parameters to the dynamic analysis of three-dimensional constrained mechanical systems. J Mech Des 104(4):785–791
4. Shabana AA (2005) Dynamics of multibody systems, 3rd edn. Cambridge University Press, Cambridge
5. Korkealaakso P, Mikkola A, Rantalainen T (2009) Description of joint constraints in the floating frame of reference formulation. Proc Inst Mech Eng Part K – J Multi-Body Dyn 223 (2):133–145
6. Bayo E, Garcia de Jalon J, Serna MA (1988) A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Comput Methods App Mech Eng 71 (2):183–195
7. Watton J (1989) Fluid fower systems. Prentice Hall International (UK) Ltd, New York, p 490
8. Handroos HM, Vilenius MJ (1991) Flexible semi-empirical models for hydraulic flow control valves. J Mech Des 113(3):232–238
9. Rouvinen A, Lehtinen T, Korkealaakso P (2005) Container gantry crane simulator for operator training. Proc Inst Mech Eng Part K – J Multi-Body Dyn 219(4):325–336
10. Korkealaakso PM, Rouvinen AJ, Moisio SM (2007) Development of a real-time simulation environment. Multibody Syst Dyn 17(2–3):177–194

# Chapter 14
# On a Momentum Based Version of Lagrange's Equations

**Hans Irschik, Michael Krommer, Manfred Nader, Yury Vetyukov, and Hans-Georg von Garssen**

*Dedicated to Franz Ziegler, Professor Emeritus of Rational Mechanics, Vienna University of Technology, on the occasion of his 75th birthday.*

**Abstract** The present contribution intends to promote an alternative form of Lagrange's Equations, which rests upon the notion of momentum. We first present a short derivation of the proposed momentum based version of Lagrange's Equations. From this derivation it becomes apparent that the derivatives of the kinetic energy with respect to the generalized coordinates must cancel out in the original kinetic energy based version of Lagrange's Equations, and thus need not to be computed. The presented momentum based formulation of Lagrange's Equations is valid for deformable bodies, modeled in the framework of the Ritz approximation technique, where rigid-body degrees-of-freedom may be present. After having stated this momentum based version of Lagrange's Equations, we restrict to plane motions of rigid bodies, and demonstrate our proposed formulation for the case of a rotational degree of freedom, where we present an additional connection to the notion of momentum of the rigid body, particularly to angular momentum. Finally, we present the exemplary application to systems consisting of two rigid bodies, namely the pendulum with a point mass and movable support, and the Sarazin pendulum consisting of a rigid rotating disc and an attached point mass.

H. Irschik (✉) • M. Krommer
Institute of Technical Mechanics, Johannes Kepler University of Linz, Linz, Austria
e-mail: hans.irschik@jku.at; michael.krommer@jku.at

M. Nader • Y. Vetyukov
Linz Center of Mechatronics GmbH (LCM), Linz, Austria
e-mail: manfred.nader@lcm.at; yury.vetyukov@lcm.at

Hans-Georg von Garssen
Siemens Corporate Technology, Munich, Germany
e-mail: hans-georg.garssen@siemens.com

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_14, 

## 14.1 Introduction

Since their discovery, Lagrange's Equations have appealed engineers and researchers from both, the application point of view, as well as from theoretical reasons. For a derivation of these relations, which has inspired the present contribution, see the book by Ziegler [13]. A collection of solved problems was presented by Gignoux and Silvestre-Brac [5]. In their original form, see [13] and [5], the dynamic terms in Lagrange's Equations refer to the notion of kinetic energy. It is the scope of the present contribution to promote an alternative form of Lagrange's Equations, which rests upon the notion of momentum. Subsequently, we first present a short derivation of the proposed momentum based version of Lagrange's Equations. From this derivation it becomes apparent that the derivatives of the kinetic energy with respect to the generalized coordinates must cancel out in the original kinetic energy based version of Lagrange's Equations, and thus need not to be computed. The presented momentum based formulation of Lagrange's Equations is valid for deformable bodies, modeled in the framework of the Ritz approximation technique, where rigid-body degrees-of-freedom may be present. After having stated this momentum based version of Lagrange's Equations, we restrict to plane motions of rigid bodies, and demonstrate our proposed formulation for the case of a rotational degree of freedom, where we present an additional connection to the notion of momentum, particularly to angular momentum, of the rigid body. Finally, we present the exemplary application to systems consisting of two rigid bodies, namely the pendulum with a point mass and movable support, and the Sarazine pendulum consisting of a rigid rotating disc and an attached rigid pendulum. A more general derivation of the proposed momentum based formulation in the framework of the Lagrange or material description of continuum mechanics, and the application to deformable bodies that move in three-dimensional space, will be presented in a forthcoming contribution [7].

## 14.2 The Original Kinetic Energy Based form of Lagrange's Equations

In the Ritz approximation technique, see Ziegler [13], the actual position vector from a space-fixed origin of some material particle of a generally deformable body is parameterized by a finite set of time-dependent generalized coordinates $q_k = q_k(t)$, $k = 1, \ldots, l$. For the sake of brevity, we take these coordinates as independent from one another, and we study the skleronomic case. Following [13], this functional dependency of the position vector is written as

$$\mathbf{r} = \mathbf{r}(q_1(t), \ldots, q_k(t), \ldots, q_l(t)). \tag{14.1}$$

The velocity of the particle becomes

$$\mathbf{v} = \sum_{k=1}^{l} \frac{\partial \mathbf{r}}{\partial q_k} \dot{q}_k, \tag{14.2}$$

where a superimposed dot stands for the derivative with respect to time. The following useful relations do hold in the present context, see Ziegler [13],

$$\frac{\partial \mathbf{r}}{\partial q_k} = \frac{\partial \mathbf{v}}{\partial \dot{q}_k}, \quad \frac{d}{dt}\left(\frac{\partial \mathbf{r}}{\partial q_k}\right) = \frac{d}{dt}\left(\frac{\partial \mathbf{v}}{\partial \dot{q}_k}\right) = \frac{\partial \mathbf{v}}{\partial q_k}. \tag{14.3}$$

As is well-known, the original kinetic energy based formulation of Lagrange's Equations, see Chap. 10 of Ziegler [13] for its derivation, reads:

$$Q_k = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_k}\right) - \frac{\partial T}{\partial q_k}; \quad k = 1, \ldots, l. \tag{14.4}$$

The $k$-th generalized force is denoted by $Q_k$. The kinetic energy $T$ follows as the integral over the mass $m$ of the body, or the system of bodies under consideration:

$$T = \int_m \frac{1}{2} \mathbf{v} \cdot \mathbf{v} \, dm. \tag{14.5}$$

The derivation of (14.4) in Chap. 10 of Ziegler [13] holds for deformable bodies with a conserved mass, where rigid-body degrees of freedom may be present. For an extension of (14.4) for deformable bodies with a time-variable mass, see Irschik and Holl [6].

## 14.3 The Proposed Momentum Based form of Lagrange's Equations

The momentum based form of Lagrange's Equations, which we propose to use, reads

$$Q_k = \int_m \frac{\partial \mathbf{v}}{\partial \dot{q}_k} \cdot \frac{d\mathbf{v}}{dt} \, dm; \quad k = 1, \ldots, l. \tag{14.6}$$

For a short-handed derivation of this formulation, it is only necessary to note that

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_k}\right) &= \int\limits_m \frac{d}{dt}\left(\frac{\partial}{\partial \dot{q}_k}\left(\frac{1}{2}\mathbf{v}\cdot\mathbf{v}\right)\right)dm = \int\limits_m \frac{d}{dt}\left(\frac{\partial \mathbf{v}}{\partial \dot{q}_k}\cdot\mathbf{v}\right)dm \\ &= \int\limits_m \frac{d}{dt}\left(\frac{\partial \mathbf{r}}{\partial q_k}\cdot\mathbf{v}\right)dm = \int\limits_m \frac{d}{dt}\left(\frac{\partial \mathbf{r}}{\partial q_k}\right)\cdot\mathbf{v}\,dm + \int\limits_m \frac{\partial \mathbf{r}}{\partial q_k}\cdot\frac{d\mathbf{v}}{dt}\,dm \\ &= \int\limits_m \frac{\partial \mathbf{v}}{\partial q_k}\cdot\mathbf{v}\,dm + \int\limits_m \frac{\partial \mathbf{v}}{\partial \dot{q}_k}\cdot\frac{d\mathbf{v}}{dt}\,dm \\ &= \frac{\partial T}{\partial q_k} + \int\limits_m \frac{\partial \mathbf{v}}{\partial \dot{q}_k}\cdot\frac{d\mathbf{v}}{dt}\,dm. \end{aligned} \tag{14.7}$$

In order to derive this result, the definition of the kinetic energy in (14.5) has been used, and the integrals over the conserved mass $m$ and the derivatives have been interchanged. Also, the derivatives with respect to time and generalized coordinates have been interchanged. The chain rule of differentiation has been applied, and (14.3) has been utilized. Substituting (14.7) into (14.4) it follows that

$$Q_k = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_k}\right) - \frac{\partial T}{\partial q_k} = \int\limits_m \frac{\partial \mathbf{v}}{\partial \dot{q}_k}\cdot\frac{d\mathbf{v}}{dt}\,dm; \quad k = 1,\ldots,l. \tag{14.8}$$

This proves (14.6). We call the latter relation a momentum based version of Lagrange's Equations, since the elementary momentum carried by a material particle is $\mathbf{v}\,dm$, and since we wish to emphasize that Lagrange's Equations are nothing else then a mathematical consequence of the Fundamental Law of Dynamics, see Ziegler [13] for the latter. The momentum based formulation of Lagrange's Equations stated in (14.6) has distinguished predecessors in the literature. For systems of mass points and rigid bodies, we mention the formalism by Kane, [10, 11] and the geometrical derivation by Casey [3, 4]. Also the projection equation proposed by Bremer in elastic multi-body dynamics, [1, 2] involves the notion of momentum. The present derivation yields a momentum based formulation in the framework of the formulation given in Chap. 10 of the book of Ziegler [13] for the kinetic energy based version, i.e. for deformable bodies, where however rigid-body degrees of freedom may be present, and where rigid bodies and mass points are included as special cases. In this sense, the momentum based version in (14.6) may be considered as a reformulation and small extension of results that were presented earlier in Refs. [1–4, 10, 11]. Also, for the sake of brevity, we above have restricted to the case of independent generalized coordinates. However, our above derivation of (14.6) clearly demonstrates that and why the derivatives of the kinetic energy $T$ with respect to the generalized coordinates $q_k$ must cancel out from the original kinetic energy based formulation, (14.4), and thus need not to be computed, see (14.7). In this sense, (14.6) represents nothing else than a compact version of

Lagrange's Equations. This will be demonstrated subsequently for the case of a plane motion of a rigid body, where a further connection to the notion of momentum will be assigned to the right hand side of (14.6).

## 14.4 Plane Motion of a Rigid Body

Consider some rigid body in a state of plane motion. In a space-fixed Cartesian coordinate system, the unit vector perpendicular to the plane of motion is denoted as $\mathbf{e}_y$, while the plane is spanned by the unit vectors $\mathbf{e}_x$ and $\mathbf{e}_z$. In order to describe the motion, we use the notation of Chap. 10 of the book by Ziegler [13]. The absolute velocity vector of a material particle in the body is written as

$$\mathbf{v} = \mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}' = \mathbf{v}_A + \left(\dot{\varphi}\mathbf{e}_y\right) \times \mathbf{r}'. \tag{14.9}$$

This is also known as Euler's formula of the kinematics of a relative motion, Ziegler [13]. The absolute velocity vector of some body-fixed reference point $A$ is $\mathbf{v}_A$, and $\mathbf{r}'$ is the relative position vector pointing from $A$ to the particle under consideration. In the present case of a rigid plane motion, the angular velocity vector of the body is $\boldsymbol{\omega} = \left(\dot{\varphi}\mathbf{e}_y\right)$, where the angle of rotation of the body from a space-fixed direction is denoted as $\varphi$.

Following Chap. 10 of Ziegler [13], the kinetic energy of the body is written as

$$T = \frac{1}{2}(\mathbf{v}_A \cdot \mathbf{v}_A)m + \left(\mathbf{v}_A \times \left(\dot{\varphi}\mathbf{e}_y\right)\right) \cdot \int_m \mathbf{r}'\, dm + \frac{1}{2}\int_m \left(\left(\dot{\varphi}\mathbf{e}_y\right) \times \mathbf{r}'\right) \cdot \left(\left(\dot{\varphi}\mathbf{e}_y\right) \times \mathbf{r}'\right) dm. \tag{14.10}$$

Subsequently, we utilize the relations

$$\int_m \mathbf{r}' dm = \mathbf{r}'_S m \tag{14.11}$$

and

$$\int_m \left(\mathbf{e}_y \times \mathbf{r}'\right) \cdot \left(\mathbf{e}_y \times \mathbf{r}'\right) dm = \left(i_S^2 + s^2\right) m, \tag{14.12}$$

where $\mathbf{r}'_S$ is the position vector pointing from $A$ to the center of mass $S$. The distance between $A$ and $S$ is denoted $s$, and $i_s$ is the radius of gyration about the center of mass, see Chap. 7 of Ziegler [13]. Note that both, $s$ and $i_S$, do not depend on time. We now take the rotation angle $\varphi$ as an independent generalized coordinate and compute the corresponding kinetic energy terms in the traditional version of Lagrange's Equations, (14.4). In doing so, we assume that $\mathbf{v}_A$ may depend on $\varphi$ and $\dot{\varphi}$, and that $\mathbf{r}'_S$ is a function of $\varphi$. We thus obtain from (14.10) the following expression

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\varphi}} = \frac{d}{dt}\left(\frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}} \cdot \mathbf{v}_A m + \left(\frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}} \times (\dot{\varphi}\mathbf{e}_y)\right) \cdot \int_m \mathbf{r}'\, dm. \right.$$
$$\left. + (\mathbf{v}_A \times \mathbf{e}_y) \cdot \mathbf{r}'_S\, m + \dot{\varphi}(i^2 + s^2)\right)$$
$$= \frac{d}{dt}\left(\frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}} \cdot \int_m (\mathbf{v}_A + (\dot{\varphi}\mathbf{e}_y) \times \mathbf{r}')dm. \right.$$
$$\left. + (\mathbf{v}_A \times \mathbf{e}_y) \cdot \mathbf{r}'_S m + \dot{\varphi}(i^2 + s^2)\right)$$
$$= \frac{d}{dt}\left(\frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}}\right) \cdot \mathbf{J} + \frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}} \cdot \frac{d}{dt}\mathbf{J} + \left(\frac{d}{dt}\mathbf{v}_A \times \mathbf{e}_y\right) \cdot \mathbf{r}'_S m$$
$$+ (\mathbf{v}_A \times \mathbf{e}_y) \cdot \frac{d}{dt}\mathbf{r}'_S m + \ddot{\varphi}(i_S^2 + s^2)m. \qquad (14.13)$$

The total momentum or impulse of the body, see Chap. 7 of Ziegler [13], is the vector

$$\mathbf{J} = \int_m \mathbf{v}\, dm = \int_m (\mathbf{v}_A + (\dot{\varphi}\mathbf{e}_y) \times \mathbf{r}')dm = (\mathbf{v}_A + (\dot{\varphi}\mathbf{e}_y) \times \mathbf{r}'_S)m. \qquad (14.14)$$

Utilizing (14.3) and noting that for the plane motion under consideration

$$\frac{d}{dt}\mathbf{r}'_S = (\dot{\varphi}\mathbf{e}_y) \times \mathbf{r}'_S, \qquad (14.15)$$

we arrive at

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\varphi}} = \frac{\partial \mathbf{v}_A}{\partial \varphi} \cdot \mathbf{J} - \dot{\varphi}(\mathbf{v}_A \cdot \mathbf{r}'_S)m + \frac{\partial \mathbf{v}_A}{\partial \dot{\varphi}} \cdot \frac{d}{dt}\mathbf{J} + \mathbf{e}_y \cdot \left(m\mathbf{r}'_S \times \frac{d}{dt}\mathbf{v}_A + \frac{d}{dt}\mathbf{H}_A\right). \qquad (14.16)$$

The total relative angular momentum of the rigid body in plane motion about $A$, see Chap. 7 of Ziegler [13] for definition, is the vector

$$\mathbf{H}_A = (\dot{\varphi}\,\mathbf{e}_y) \int_m \mathbf{r}' \cdot \mathbf{r}'\, dm = (\dot{\varphi}\mathbf{e}_y)(i_S^2 + s^2)m. \qquad (14.17)$$

Proceeding with the kinetic energy based version of Lagrange's Equations, (14.4), we obtain from (14.10) that

$$\begin{aligned}\frac{\partial T}{\partial \varphi} &= \left(\frac{\partial \mathbf{v}_A}{\partial \varphi}\cdot \mathbf{v}_A + \left(\frac{\partial \mathbf{v}_A}{\partial \varphi}\times\left(\dot\varphi \mathbf{e}_y\right)\right)\cdot \mathbf{r}'_S + \left(\mathbf{v}_A\times\left(\dot\varphi\mathbf{e}_y\right)\right)\cdot\frac{\partial \mathbf{r}'_S}{\partial\varphi}\right)m\\ &= \frac{\partial \mathbf{v}_A}{\partial\varphi}\cdot\int_m \left(\mathbf{v}_A + \left(\dot\varphi\mathbf{e}_y\right)\times\mathbf{r}'\right)dm + \left(\mathbf{v}_A\times\left(\dot\varphi\mathbf{e}_y\right)\right)\cdot\left(\mathbf{e}_y\times\mathbf{r}'_S\right)m\\ &= \frac{\partial \mathbf{v}_A}{\partial\varphi}\cdot\mathbf{J} - \dot\varphi\left(\mathbf{v}_A\cdot\mathbf{r}'_S\right)m, \end{aligned} \tag{14.18}$$

where we have used (14.14) and the fact that in a rigid plane motion

$$\frac{\partial \mathbf{r}'_S}{\partial\varphi} = \mathbf{e}_y\times\mathbf{r}'_S\,. \tag{14.19}$$

Substituting (14.18) in (14.16), we find that

$$\frac{d}{dt}\frac{\partial T}{\partial\dot\varphi} = \frac{\partial T}{\partial\varphi} + \frac{\partial \mathbf{v}_A}{\partial\dot\varphi}\cdot\frac{d}{dt}\mathbf{J} + \mathbf{e}_y\cdot\left(m\mathbf{r}'_S\times\frac{d}{dt}\mathbf{v}_A + \frac{d}{dt}\mathbf{H}_A\right). \tag{14.20}$$

This again gives evidence for the fact that the derivative of the kinetic energy with respect to the generalized coordinate, in the present case (14.18), does cancel out from Lagrange's Equations, (14.4), and needs not to be computed, see (14.7) for a more general setting. However, (14.20) also demonstrates that Lagrange's Equations can be assigned with a clear mechanical meaning in terms of the notions of momentum and relative angular momentum of the body as a whole. Indeed, from (14.9) we obtain

$$\frac{\partial \mathbf{v}}{\partial\dot\varphi} = \frac{\partial \mathbf{v}_A}{\partial\dot\varphi} + \mathbf{e}_y\times\mathbf{r}'. \tag{14.21}$$

The proposed momentum based version of Lagrange's Equations, (14.6), thus yields

$$\begin{aligned}\int_m \frac{\partial \mathbf{v}}{\partial\dot\varphi}\cdot\frac{d}{dt}\mathbf{v}\,dm &= \frac{\partial \mathbf{v}_A}{\partial\dot\varphi}\cdot\frac{d}{dt}\mathbf{J} + \int_m\left(\mathbf{e}_y\times\mathbf{r}'\right)\cdot\frac{d}{dt}\mathbf{v}\,dm\\ &= \frac{\partial \mathbf{v}_A}{\partial\dot\varphi}\cdot\frac{d}{dt}\mathbf{J} + \mathbf{e}_y\cdot\int_m \mathbf{r}'\times\frac{d}{dt}\mathbf{v}\,dm, \end{aligned} \tag{14.22}$$

where we have used (14.14). Using (14.9), there is

$$\int_m \mathbf{r}'\times\frac{d}{dt}\mathbf{v}\,dm = m\mathbf{r}'_S\times\frac{d}{dt}\mathbf{v}_A + \int_m \mathbf{r}'\times\frac{d}{dt}\left(\left(\dot\varphi\mathbf{e}_y\right)\times\mathbf{r}'\right)dm. \tag{14.23}$$

In accordance with Chap. 1 of Ziegler [13], the kinematics of a relative motion yields

$$\frac{d}{dt}\left(\left(\dot{\varphi}\mathbf{e}_y\right)\times\mathbf{r}'\right)=\left(\ddot{\varphi}\mathbf{e}_y\right)\times\mathbf{r}'+\dot{\varphi}^2\left(\left(\mathbf{e}_y\cdot\mathbf{r}'\right)\mathbf{e}_y-\mathbf{r}'\right). \tag{14.24}$$

Thus, using (14.17), we obtain

$$\int_m \mathbf{r}'\times\frac{d}{dt}\left(\left(\dot{\varphi}\mathbf{e}_y\right)\times\mathbf{r}'\right)dm=\left(\ddot{\varphi}\mathbf{e}_y\right)\int_m \mathbf{r}'\cdot\mathbf{r}'\,dm=\left(\ddot{\varphi}\mathbf{e}_y\right)\left(i_S^2+s^2\right)m=\frac{d}{dt}\mathbf{H}_A. \tag{14.25}$$

Substituting (14.25) into (14.22) yields

$$\int_m \frac{\partial\mathbf{v}}{\partial\dot{\varphi}}\cdot\frac{d}{dt}\mathbf{v}\,dm=\frac{\partial\mathbf{v}_A}{\partial\dot{\varphi}}\cdot\frac{d}{dt}\mathbf{J}+\mathbf{e}_y\cdot\left(m\mathbf{r}'_S\times\frac{d}{dt}\mathbf{v}_A+\frac{d}{dt}\mathbf{H}_A\right), \tag{14.26}$$

which, when compared with (14.20), gives evidence for the correctness of (14.8). Compare also the three-dimensional result that was given by Casey [4] for the case that $A$ is the center of mass, and the velocity of $A$ does not depend on the rotational speed.

For the sake of comparison with two subsequent benchmark examples, we write out (14.26) in terms of the rotational generalized coordinate $\varphi$. Substituting

$$\mathbf{v}_A=\dot{x}_A\left(\varphi,\,\dot{\varphi}\right)\mathbf{e}_x+\dot{z}_A\left(\varphi,\,\dot{\varphi}\right)\mathbf{e}_z \tag{14.27}$$

and

$$\mathbf{r}'_S=s(\,\sin\varphi\mathbf{e}_x+\cos\varphi\mathbf{e}_z) \tag{14.28}$$

into (14.26) and using (14.14) and (14.17), we eventually obtain

$$\begin{aligned}\int_m \frac{\partial\mathbf{v}}{\partial\dot{\varphi}}\cdot\frac{d}{dt}\mathbf{v}\,dm=\Bigg(&\frac{\partial\dot{x}_A}{\partial\dot{\varphi}}\left(\ddot{x}_A+s\left(\ddot{\varphi}\cos\varphi-\dot{\varphi}^2\sin\varphi\right)\right)\\&+\frac{\partial\dot{z}_A}{\partial\dot{\varphi}}\left(\ddot{z}_A-s\left(\ddot{\varphi}\sin\varphi+\dot{\varphi}^2\cos\varphi\right)\right)\\&+s(\ddot{x}_A\cos\varphi-\ddot{z}_A\sin\varphi)+\ddot{\varphi}\left(i_S^2+s^2\right)\Bigg)m.\end{aligned} \tag{14.29}$$

Note that, in deriving this result, no terms appear to cancel out.

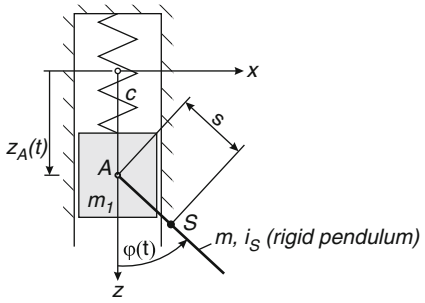



**Fig. 14.1** Vertical pendulum

## 14.5 Plane Pendulum with a Movable Support

As a first example, we consider the case of a plane pendulum with a vertically moving support, treated in Sect. 10.2.2 of Ziegler [13]. The system consists of two rigid bodies, a point mass $m_1$, which can perform a translational motion in the direction of $\mathbf{e}_z$ only, and a rigid pendulum with mass $m$, the latter being linked to the point mass at the point $A$, see Fig. 14.1. This system has two independent degrees of freedom, the vertical displacement $z_A$ of the link $A$, and the rotation angle $\varphi$ of the pendulum. Since the movement of the support $A$ in the direction of $\mathbf{e}_x$ is constrained, $x_A = 0$, and since $z_A$ is an independent degree of freedom, there is

$$\frac{\partial \dot{x}_A}{\partial \dot{\varphi}} = 0, \quad \ddot{x}_A = 0, \quad \frac{\partial \dot{z}_A}{\partial \dot{\varphi}} = 0. \tag{14.30}$$

The problem was treated by the original kinetic energy based version of Lagrange's equations in Sect. 10.2.2 of Ziegler [13], where the following results were obtained:

$$\frac{\partial T}{\partial \varphi} = -ms\dot{z}_A\,\dot{\varphi}\cos\varphi \tag{14.31}$$

and

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\varphi}}\right) = m\left(s^2 + i_S^2\right)\ddot{\varphi} - ms(\ddot{z}_A \sin\varphi + \dot{z}_A\dot{\varphi}\cos\varphi). \tag{14.32}$$

Note that the notation $z_A = z_q$ was used in Sect. 10.2.2 of Ziegler [13]. Substituting (14.31) and (14.32) into the original kinetic energy based version of Lagrange's equations, (14.4), it is seen that (14.31) indeed does cancel out. The result is:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\varphi}}\right) - \frac{\partial T}{\partial \varphi} = -ms\ddot{z}_A \sin\varphi + m\left(s^2 + i_S^2\right)\ddot{\varphi}. \tag{14.33}$$

We now proceed to the proposed momentum based version, (14.29). The momentum based scalar quantities considered in the latter are additive, like the kinetic energy terms in the original version used in [13], such that the momentum based quantities in (14.29) can be computed separately for the single bodies under consideration, and can be added afterwards. For the point mass $m_1$, there is no rotation to be considered, such that the point mass does not contribute to (14.29). Hence, it is sufficient to take into account the pendulum only. Substituting (14.30) into (14.29), we directly obtain Ziegler's result, (14.33). This gives evidence for the correctness of (14.29).

## 14.6 The Sarazin Pendulum in Plane Motion

As a second example, we consider the case of a Sarazin pendulum, treated in Sect. 10.7, exercise A 10.3 of Ziegler [13]. The system consists of two rigid bodies, a disc with mass $m_1$, rotating about its space-fixed center of mass, and a rigid pendulum with mass $m_2$, the pendulum being linked to the disc at the point $A$. Both bodies move in the same plane. The system has two independent degrees of freedom. In order to be compatible with (14.29), we choose the absolute rotation angle $\varphi$ of the pendulum, and the relative rotation angle $\psi$ between the two bodies, see Fig. 14.2. Note that the absolute rotation angle of the disc and the relative rotation angle were used as degrees of freedom in Sect. 10.7, exercise A 10.3 of Ziegler [13].

From Fig. 14.2 it is seen that

$$x_A = R\sin(\varphi - \psi), \quad z_A = R\cos(\varphi - \psi) \tag{14.34}$$

such that

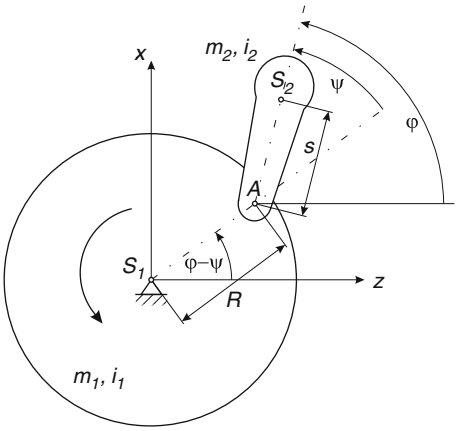
$$\frac{\partial \dot{x}_A}{\partial \dot{\varphi}} = R\cos(\varphi - \psi), \quad \frac{\partial \dot{z}_A}{\partial \dot{\varphi}} = -R\sin(\varphi - \psi) \tag{14.35}$$

and

$$\begin{aligned} \ddot{x}_A &= R\left(\ddot{\varphi} - \ddot{\psi}\right)\cos(\varphi - \psi) - R\left(\dot{\varphi} - \dot{\psi}\right)^2 \sin(\varphi - \psi), \\ \ddot{z}_A &= -R\left(\ddot{\varphi} - \ddot{\psi}\right)\sin(\varphi - \psi) - R\left(\dot{\varphi} - \dot{\psi}\right)^2 \cos(\varphi - \psi). \end{aligned} \tag{14.36}$$

Applying (14.29) separately for the two bodies under consideration and adding the results, where (14.34) and (14.36) are substituted for the pendulum, we obtain

**Fig. 14.2** Sarazin pendulum




$$\int_m \frac{\partial \mathbf{v}}{\partial \dot{\varphi}} \cdot \frac{d}{dt}\mathbf{v}\, dm = m_1 i_1^2 \left(\ddot{\varphi} - \ddot{\psi}\right) + m_2 \Big(R^2 \left(\ddot{\varphi} - \ddot{\psi}\right) - sR\left(2\dot{\varphi}\dot{\psi} - \dot{\psi}^2\right)\sin\psi$$
$$+sR\left(2\ddot{\varphi} - \ddot{\psi}\right)\cos\psi + \ddot{\varphi}\left(i_2^2 + s^2\right)\Big). \tag{14.37}$$

An equivalent expression was given in Sect. 10.7, exercise A 10.3 of Ziegler [13], as a result of the original kinetic energy based version of Lagrange's equations. (Due to the different meaning of $\varphi$, our $\varphi$ must be replaced by $(\varphi + \psi)$ in (14.37) in order to obtain Ziegler's expression.) This equivalence again gives evidence for the correctness of (14.29).

## 14.7 Conclusion

In Sect. 14.3 above, the proposed momentum based version of Lagrange's Equations, (14.6), has been derived from the original kinetic energy based version, (14.4). The derivation, see (14.7), demonstrates that and why the derivatives of the kinetic energy with respect to the generalized coordinates must cancel out from the kinetic energy based version of Lagrange's Equations. The presented momentum based formulation of Lagrange's Equations is valid for deformable bodies, modeled in the framework of the Ritz approximation technique, where rigid-body degrees-of-freedom may be present. In Sect. 14.4 above, the formulation has been worked out for rigid bodies in plane motion, where an additional mechanical meaning has been attached in terms of total momentum and total relative angular momentum of the rigid bodies, (14.26), see (14.29) for an explicit formulation. Finally, the application to two systems consisting of two rigid bodies has been presented, namely the pendulum with a point mass and movable support, and the Sarazin pendulum consisting of a rigid rotating disc and an attached point mass, see

Sects. 14.5 and 14.6. The question, why certain terms do cancel out in the original kinetic energy based version of Lagrange's Equations became apparent to the senior author when working as a young assistant of Professor Franz Ziegler. An industrial research project, performed in the framework of ACCM and concerning rapidly rotating deformable bodies, see [8, 9, 12], recently has given the opportunity to the present authors to reconsider this question and to work out the above solution. A more general presentation of the proposed momentum based formulation in the framework of the Lagrange or material description of continuum mechanics, and the application to deformable bodies that move in three-dimensional space, will be presented in a forthcoming contribution [7].

**Acknowledgements** Support of the present work in the framework of the COMET-K2 Austrian Center of Competence in Mechatronics, ACCM, is gratefully acknowledged.

## References

1. Bremer H (2008) Elastic multibody dynamics: a direct Ritz approach. Springer, Berlin
2. Bremer H (2010) Problems in fast moving non-holonomic elastic systems. In: Mechanics and model-based control of smart materials and structures. Springer, Wien/New York, pp 19–28
3. Casey J (1994) Geometrical derivation of Lagranges equations for a system of particles. Am J Phy 62:836–847
4. Casey J (1995) On the advantages of a geometrical viewpoint in the derivation of Lagranges equations for a rigid continuum. Z Angew Math Phys (ZAMP) 46:805–847
5. Gignoux C, Silvestre-Brac B (2009) Solved problems in Lagrangian and Hamiltonian mechanics. Springer, Dordrecht
6. Irschik H, Holl H (2002) The equations of Lagrange written for a non-material volume. Acta Mech 153:231–248
7. Irschik H, Krommer M, Nader M, Vetyukov Y, von Garssen H-G The equations of lagrange for a continuous deformable body with rigid body degrees of freedom, written in a momentum based formulation. Forthcoming
8. Irschik H, Nader M, Stangl M, von Garssen H-G (2009) A floating frame-of-reference formulation for deformable rotors using the properties of free elastic vibration modes. In: Proceedings of the ASME 2009 international design engineering technical conferences and computers and information in engineering conference IDETC/CIE, San Diego, no. ASME-Paper No. DETC2009-86660, p 8
9. Irschik H, Nader M, Stangl M, von Garssen H-G (2011) A model reduction technique for high speed flexible rotors. In: Advanced dynamics and model-based control of structures and machines. Springer, Wien/New York, pp 127–134
10. Kane T (1961) Dynamics of nonholonomic systems. J App Mech 83:574–578
11. Kane T, Levinson DA (1985) Dynamics: theory and applications. Mc-GrawHill, New York
12. Nader M, Irschik H, Stangl M, von Garssen H-G (2010) Nonlinear vibrations of flexible high-speed rotors supported by visco-elastic bearings. In: Proceedings of the 8th IFToMM international conference on rotordynamics, Seoul, pp 1054–1061
13. Ziegler F (1998) Mechanics of solids and fluids, Corrected reprint of 2nd edn. Springer, New York, Vienna

# Chapter 15
# Vibration Control and Structural Damping of a Rotating Beam by Using Piezoelectric Actuators

**Christian Zehetner and Georg Zenz**

**Abstract** In this paper, the application of piezoelectric vibration control in flexible multibody systems is studied and verified. Exemplarily, beam-type structures are considered that are subject to inertial and external forces. The equations of motion for three-dimensional flexible and torsional vibrations are presented considering the influence of piezoelectric actuation strains. In the framework of Bernoulli-Euler beam theory the shape control solution is derived, i.e. the distribution of actuation strains such that the flexible displacements are completely compensated. For the experimental verification, a laboratory model has been developed, in which the theoretical distribution of actuation strains is discretized by piezoelectric patches. A suitable control algorithm is implemented within a dSpace environment. Finally, the results are validated by numerical computations utilizing *ABAQUS* and *HOTINT*, and verified by experimental evaluation.

## 15.1 Introduction

Recently, the interest in vibration compensation by means of distributed actuation has increased rapidly. On the one hand, structures become more and more light-weighted, on the other hand there are considerable advances in the development of materials suitable for such kinds of actuators and sensors. This paper concentrates on the application of piezoelectric transducers in order to control flexible vibrations in beams, which are important components of many multibody systems.

Piezoelectric transducers can be used for sensing and actuation, utilizing either the direct or the converse piezoelectric effect, respectively [1]. An efficient possibility for realisation is to apply piezoelectric patches on the surfaces of beams.

C. Zehetner (✉) • G. Zenz
Austrian Center of Competence in Mechatronics (ACCM), Altenbergerstr. 69, 4040 Linz, Austria
e-mail: christian.zehetner@accm.co.at; georg.zenz@accm.co.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_15, 

Depending on the type of the piezoelectric material, and on the position of the patch on the beam, such transducers can be used for sensing and actuation of bending and torsional modes. Vibration compensation by piezoelectric materials has been extensively treated in the literature. The exact compensation of flexible displacements by distributed actuation has been denoted as shape control, for a review see Irschik [2].

An exact solution in the framework of Bernoulli-Euler beam theory for the complete compensation of plane bending vibrations under influence of rigidbody motions has been presented by Zehetner and Irschik [3]. Torsional vibration control has been investigated by Zehetner and Krommer [4], where it has been shown how piezoelectric transducers can be used for torsional sensing and actuation. A comparison of some specific piezoelectric materials for the application of torsional actuation and sensing has been shown in [5].

There are several possibilities for the practical realisation of the spatial distribution, i.e. the shape of the actuators and sensors. For instance, shaped piezoelectric layers can be applied on the beam. Other possibilities would be shaped electrodes or functionally graded material properties. These strategies enable the exact distribution of the necessary actuation strains, but are very extensive. Thus, patch approximations are more suitable for practical applications. A patch approximation for the control of vibrations of a rotating beam has been investigated numerically by Zehetner and Gerstmayr [6], and first experimental results have been presented in [7] and [8].

Goal of this work is the derivation and verification of a mechanical model for the control of three-dimensional flexural and torsional beam vibrations caused by external forces and inertial forces due to rigidbody motions. The theoretical results are validated by numerical computations using the finite element software *ABAQUS* and the multibody dynamics simulation code *HOTINT*, mainly developed by Gerstmayr [9].

Finally, the theoretical results are verified by experimental investigations. For this sake, a laboratory model has been set up, in which 48 piezoelectric patches are applied on a rectangular hollow beam. The beam is fixed on a motor, such that it performs a rotational rigid body motion. Within a *dSpace* environment control algorithms are implemented and tested. It turns out that the flexible vibrations can be reduced significantly, and that theoretical, numerical and experimental results show a very good coincidence.

## 15.2 Piezoelectric Actuators and Sensors

Piezoelectric layers can be used as actuators and sensors in various ways. Here, we consider piezoelectric patches that are attached on the surfaces of a beam. We distinguish between two operational modes:

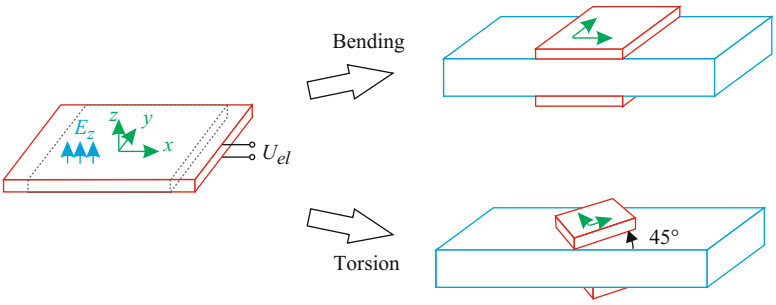



**Fig. 15.1** Piezoelectric extension mode

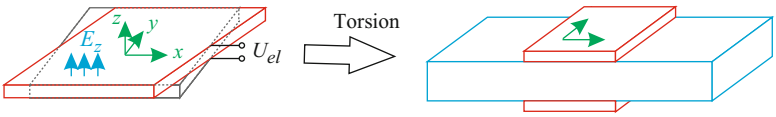



**Fig. 15.2** Piezoelectric shear mode

- **Extension mode**. The electric field component in thickness direction corresponds to extension strains as shown in Fig. 15.1. Bending actuators and sensors are realized by placing such devices symmetrically on the upper and lower surface of the beam and applying an electric field in opposite direction (extension and contraction). Such a behaviour is provided e.g. by the piezoelectric ceramic PZT (lead zirconate titanate). Extension with a predominating axis can be realized by makro-fiber composites (MFC) consisting of PZT stripes embedded in epoxy-substrate. Torsional actuation and sensing can be realized by placing such layers at an angle of 45° with respect to the rod axis as shown in Fig. 15.1.
- **Shear mode**. The electric field in thickness direction corresponds to shear strains as shown in Fig. 15.2. Such a behaviour is shown e.g. by the piezoelectric material ADP (ammonium dihydrogen phosphate). The shear mode can be utilized for torsional sensing and actuation as shown in Fig. 15.2.

## 15.3 Constitutive Equations

The constitutive equations for piezoelectric materials relate the mechanical strain $\boldsymbol{\varepsilon} = [\varepsilon_{xx}\ \varepsilon_{yy}\ \varepsilon_{zz}\ \gamma_{yz}\ \gamma_{xz}\ \gamma_{xy}]^T$, stress $\boldsymbol{\sigma} = [\sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \tau_{yz}\ \tau_{xz}\ \tau_{xy}]^T$, electrical field $\boldsymbol{E} = [E_x\ E_y\ E_z]^T$ and dielectric displacement $\boldsymbol{D} = [D_x\ D_y\ D_z]^T$, cf. Tauchert [10], in the form

$$\boldsymbol{\sigma} = \boldsymbol{Q} \cdot (\varepsilon - \boldsymbol{dE}), \tag{15.1}$$

$$\boldsymbol{D} = \boldsymbol{\eta} \cdot \boldsymbol{E} + \boldsymbol{d}\boldsymbol{\sigma}, \tag{15.2}$$

where $\boldsymbol{Q}$ is the $6 \times 6$ matrix of elasticity coefficients, $\boldsymbol{d}$ the $6 \times 3$ matrix of piezoelectric coefficients and $\boldsymbol{\eta}$ the $3 \times 3$ matrix of dielectric coefficients. The coefficients of the matrices depend on the specific type of piezoelectric material. Examples for PZT, ADP and MFC are summarized in the Appendix.

In beam-type structures it is assumed that the stress components $\sigma_{yy}$, $\sigma_{zz}$ and $\tau_{yz}$ can be neglected, such that (15.1) and (15.2) reduce to

$$\begin{aligned} \sigma_{xx} &= \overline{Q}_{11}\,(\varepsilon_{xx} - \varepsilon^0_{xx}), \\ \tau_{xz} &= Q_{55}(\gamma_{xz} - \gamma^0_{xz}), \\ \tau_{xy} &= Q_{66}(\gamma_{xy} - \gamma^0_{xy}), \end{aligned} \tag{15.3}$$

and

$$D_z = \eta_{33}(E_z - E^0_z), \tag{15.4}$$

where $\overline{Q}_{11} = S^{-1}_{11}$ is the effective Young modulus, $S_{11}$ is the first component of the compliance matrix $\boldsymbol{S} = \boldsymbol{Q}^{-1}$. In (15.3), $\varepsilon^0_{xx}$, $\gamma^0_{xz}$ and $\gamma^0_{xy}$ are piezoelectric eigenstrains representing the converse piezoelectric effect, cf. Mura [11] for the definition of eigenstrains. Accordingly, $E^0_z$ is the electric eigenfield, a generalized formulation for the direct piezoelectric effect which has been introduced by Irschik et al. [12]. Eigenstrains and the eigenfield depend on the material properties, e.g. for PZT there is

$$\varepsilon^0_{xx} = d_{31}E_z,\ \gamma^0_{xy} = \gamma^0_{xz} = 0, \quad E^0_z = \frac{d_{31}}{\eta_{33}}\sigma_{xx}, \tag{15.5}$$

and for ADP

$$\varepsilon^0_{xx} = 0,\ \gamma^0_{xy} = d_{36}E_z,\ \gamma^0_{xz} = 0, \quad E^0_z = \frac{d_{36}}{\eta_{33}}\tau_{xy}. \tag{15.6}$$

Assuming that the piezoelectric transducers are relatively thin, the electric field components in the plane of the layers are neglected $E_x = E_y = 0$, and the electric field is constant in thickness direction and proportional to the applied voltage $U_{el}$,

$$E_z = \frac{U_{el}}{h_L}, \tag{15.7}$$

$h_L$ is the thickness of the piezoelectric layer.

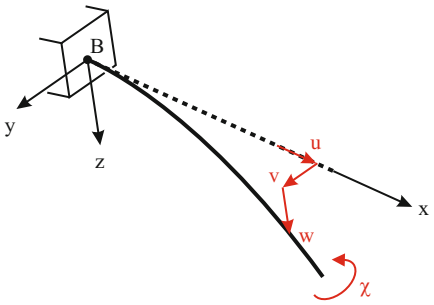



**Fig. 15.3** Moving cantilever beam, ideally fixed on a rigid base

## 15.4 Kinematics

Figure 15.3 shows a flexible beam fixed to a rigid base in point B. A floating reference frame is introduced with origin in point B. In the undeformed configuration, $x$ coincides with the beam axis, and $(y, z)$ is the cross-sectional plane. The position of B with respect to an inertial frame is described by the position vector $\boldsymbol{x}_B(t)$, and the orientation of the floating frame by the rotation matrix $\boldsymbol{A}_B(t)$. Hence, $\boldsymbol{x}_B$ and $\boldsymbol{A}_B$ represent the rigidbody motion of the beam.

The deformed beam axis is given by the flexible displacements $u(x,t)$, $v(x,t)$, $w(x,t)$ as shown in Fig. 15.3. According to Bernoulli-Euler beam theory the cross-sections remain undeformed and perpendicular to the beam axis. According to Saint Venant's theory of torsion, the cross-section performs a rigidbody rotation around the beam axis with the torsional angle $\chi(x,t)$, and an axial displacement (cross-sectional warping) expressed by Saint Venant's warping function $\varphi(y,z)$. It has been shown by Zehetner [13] that eigenstrains cause an additional cross-sectional warping which can be formulated by the warping function $\phi^0(y,z,\varepsilon^0(t))$. Thus, the displacement field of the beam is expressed by

$$\boldsymbol{u} = \begin{bmatrix} u - yv' - zw' + \chi'\varphi \\ v - z\chi \\ w + y\chi \end{bmatrix} + \begin{bmatrix} \phi^0 \\ 0 \\ 0 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 2\chi(yw' - zv') \\ zv'w' + y(v'^2 + \chi^2) \\ yv'w' + z(w'^2 + \chi^2) \end{bmatrix}. \tag{15.8}$$

The first term represents the displacements according to linear Bernoulli-Euler beam theory and Saint Venant's theory of torsion. The second term stands for the additional cross-sectional warping due to eigenstrains, and the third term contains second order terms which enable the consideration of dynamic stiffening effects and stability investigations.

For laminated cross-sections as shown in Figs. 15.1 and 15.2, the Saint Venant warping function $\varphi(y,z)$ is given by the boundary value problem

$$
\begin{aligned}
Q_{66}\frac{\partial^2\varphi}{\partial y^2}+Q_{55}\frac{\partial^2\varphi}{\partial z^2}&=0,\\
\partial A:\quad Q_{66}\left(\frac{\partial\varphi}{\partial y}-z\right)n_y+Q_{55}\left(\frac{\partial\varphi}{\partial z}+y\right)n_z&=0,\\
\partial I:\quad \left[\left[Q_{66}\left(\frac{\partial\varphi}{\partial y}-z\right)n_y+Q_{55}\left(\frac{\partial\varphi}{\partial z}+y\right)n_z\right]\right]&=0,\\
[\![\varphi]\!]&=0,
\end{aligned}
\tag{15.9}
$$

$ny$ and $nz$ are the components of the outer normal vector. A derivation can be found e.g. in Rand and Rovenski [14]. Besides the boundary conditions at the boundary $\partial A$ of the cross-section, interface conditions have to be satisfied at the interface $\partial I$ between two layers in order to obtain continuous displacement and stress distributions. In this context, the notation $[\![\cdot]\!]$ stands for the difference of a quantity at the interface.

The additional warping function $\phi^0(y,z,\boldsymbol{\varepsilon}^0(t))$ is expressed by a similar boundary problem

$$
\begin{aligned}
Q_{66}\frac{\partial^2\phi^0}{\partial y^2}+Q_{55}\frac{\partial^2\phi^0}{\partial z^2}&=0,\\
\partial A:\quad Q_{66}\left(\frac{\partial\phi^0}{\partial y}-\gamma_{xy}^0\right)n_y+Q_{55}\left(\frac{\partial\phi^0}{\partial z}-\gamma_{xz}^0\right)n_z&=0,\\
\partial I:\quad \left[\left[Q_{66}\left(\frac{\partial\phi^0}{\partial y}-\gamma_{xy}^0\right)n_y+Q_{55}\left(\frac{\partial\phi^0}{\partial z}-\gamma_{xz}^0\right)n_z\right]\right]&=0,\\
[\![\phi^0]\!]&=0,
\end{aligned}
\tag{15.10}
$$

the derivation as well as an analytical solution for rectangular laminated cross-sections can be found in Zehetner [13]. Note that (15.9) and (15.10) hold for any kind of laminated cross-sections and material behaviour according to Sect. 15.3.

## 15.5 Equations of Motion

The equations of motion for the beam in Fig. 15.3 can be derived e.g. by applying D'Alembert's principle. A detailed derivation can be found in Zehetner [15]. As excitations we consider inertial forces due to rigidbody motions as well as distributed and concentrated external forces. With the kinematical assumptions in (15.8) and the constitutive equations in (15.3) we obtain the equations of motion for longitudinal, transversal and torsional beam vibrations

$$
\begin{aligned}
f_x^e - N^{a\prime} &= \mu(\ddot{u} + a_1) - (A_{11}u')', \\
f_y^e + M_z^{a\prime\prime} &= \mu(\ddot{v} + a_2) + (D_{22}v'')'' - (Nv' - \tfrac{1}{2}M_x w'')' + (M_y\chi + \tfrac{1}{2}M_x w')'', \\
f_z^e - M_y^{a\prime\prime} &= \mu(\ddot{w} + a_3) + (D_{11}w'')'' - (Nw' + \tfrac{1}{2}M_x v'')' + (M_z\chi - \tfrac{1}{2}M_x v')'', \\
m_x^e - M_x^{a\prime} &= I_x(\ddot{\chi} + \alpha_1) - (C_{11}\chi')' + M_z w'' + M_y v'',
\end{aligned} \tag{15.11}
$$

with kinematic boundary conditions for the clamped end,

$$
x = 0: \quad u = v = v' = w = w' = \chi = \chi' = 0, \tag{15.12}
$$

and dynamic boundary conditions at the free end,

$$
\begin{aligned}
x = L: \quad F_x^e + N^a &= A_{11}u', \\
F_y^e - M_z^{a\prime} &= -(D_{22}v'')' + Nv' - \tfrac{1}{2}M_x^e w'' - (M_y\chi + \tfrac{1}{2}M_x w')', \\
M_z^e + M_z^a &= D_{22}v'' + M_y\chi + \tfrac{1}{2}M_x w', \\
F_z^e + M_y^{a\prime} &= -(D_{11}w'')' + Nw' + \tfrac{1}{2}M_x v'' - (M_z\chi - \tfrac{1}{2}M_x v')', \\
M_y^e + M_y^a &= -D_{11}w'' - M_z\chi + \tfrac{1}{2}M_x v', \\
M_x^e + M_x^a &= C_{11}\chi'.
\end{aligned} \tag{15.13}
$$

In (15.11)–(15.13), $f_x^e, f_y^e, f_z^e$ and $m_x^e$ are effective distributed external forces and torque per unit length, respectively. These effective quantities consider external forces and inertial forces due to the rigidbody motion. $F_x^e$, $F_y^e$, $F_z^e$, $M_x^e$, $M_y^e$ and $M_z^e$ are external concentrated forces acting at the free beam end, e.g. joint forces or manipulator forces. $\mu$ is the mass per unit length, $A_{11}$ the longitudinal stiffness, $D_{11}$ and $D_{22}$ are the bending stiffnesses, and $C_{11}$ is the torsional stiffness. $a_1$, $a_2$, $a_3$ and $\alpha_1$ are accelerations corresponding to the flexible displacements $u$, $v$, $w$ and $\chi$, for details see Ref. [15].

The influence of the piezoelectric effect is represented by the actuating force and moments $N^a$, $M_z^a$, $M_y^a$ and $M_x^a$. Using the piezoelectric material PZT we obtain

$$
\begin{aligned}
N^a &= \frac{U_{el}}{h_L}\int_A \overline{Q}_{11}\, d_{31}\, dA, \\
M_z^a &= \frac{U_{el}}{h_L}\int_A \overline{Q}_{11}\, d_{31} y\, dA, \\
M_y^a &= \frac{U_{el}}{h_L}\int_A \overline{Q}_{11}\, d_{31} z\, dA.
\end{aligned} \tag{15.14}
$$

On the other hand, using patches made of the material ADP, and using the substitution $\phi^0 = \overline{\phi}^0\, U_{el}$, we obtain the actuating torque

$$
M_x^a = -\frac{U_{el}}{h_L}\left(\int_{A_p} Q_{66} d_{36} z\, dA - \int_A \left(Q_{55}\frac{\partial\,\overline{\phi}^0}{\partial z} y - Q_{66}\frac{\partial\,\overline{\phi}^0}{\partial y} z\right) dA\right). \tag{15.15}
$$

## 15.6 Shape Control

From (15.11)–(15.13) we can immediately derive relations for the actuating forces and moments in order to compensate the external excitations (shape control), i.e. homogenous equations of motion are obtained if the left hand sides of (15.11)–(15.13) vanish, hence

$$\begin{aligned} & N^{a\prime} = f_x^e,\ M_z^{a\prime\prime} = -f_y^e,\ M_y^{a\prime\prime} = f_z^e,\ M_x^{a\prime} = m_x^e \\ x = L: \quad & N^a = -F_x^e,\ M_z^{a\prime} = F_y^e,\ M_z^a = -M_z^e, \\ & M_y^{a\prime} = -F_z^e,\ M_y^a = -M_y^e,\ M_x^a = -M_x^a. \end{aligned} \tag{15.16}$$

Integrating (15.16) yields the spatial distribution of actuating forces and moments in order to compensate external excitations and the influence of the rigidbody accelerations. If the motion starts from rest (homogenous initial conditions), and if no buckling effects occur, then the elastic displacements are compensated exactly. Note that buckling phenomena can also be investigated since second order terms are considered in the equations of motion.

A common strategy for the practical realisation of distributed actuation is an approximation by a patch discretisation [16]. The latter will be discussed in more detail by means of the examples in the subsequent section.

## 15.7 Examples

In order to verify the theoretical results of the above sections, two examples are considered. First, a flexible manipulator is studied numerically, and secondly, numerical and experimental results concerning a rotating beam are presented.

### *15.7.1 Flexible Manipulator—Numerical Simulations*

Figure 15.4 shows a flexible manipulator: A flexible beam is fixed on a rigid base moving in $z$-direction with the acceleration $a_3$. At the free beam end, the mass $M$ is fixed with its center of gravity located at a distance of $r$ with respect to the beam axis. The effective excitations due to the acceleration $a_3$ are represented by

$$f_z^e = -\mu a_3(t), \qquad F_z^e = -M a_3(t). \tag{15.17}$$

In the following, it is assumed that the influence of the tip mass $M$ dominates, such that the distributed force $f_z^e$ is neglected. Moreover, longitudinal and transversal vibrations in $y$-direction are neglected. Due to the force $F_z^e$, transversal vibrations

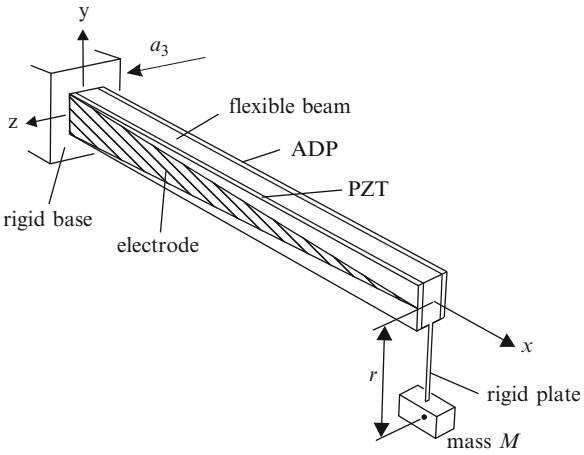



**Fig. 15.4** Flexible manipulator

in $z$-direction and torsional vibrations are excited. Inserting (15.17) into (15.16) and integrating yields the actuating bending moment $M_y^a$ and the actuating torque $M_x^a$ as

$$M_y^a(x,t) = -LMa_3(t)\left(1 - \frac{x}{L}\right), \quad M_x^a(x,t) = -Mra_3(t). \tag{15.18}$$

The spatial distribution of $M_y^a$ is linear with respect to the longitudinal coordinate $x$, and $M_x^a$ is constant. The realisation of actuation is shown in Fig. 15.4. Two piezoelectric layers are bonded ideally on the surfaces of the beam: One layer made of ADP is placed on the back side of the beam, using an electrode with constant width. On the second side, a PZT layer is attached. The width of the electrode corresponds to the linear spatial distribution of the actuating moment $M_y^a$ in (15.18). The voltage of the actuators is obtained from (15.14) and (15.15).

In order to verify (15.18), a Finite Element model has been implemented using *ABAQUS*. The beam and the piezoelectric layers have been discretized by 3D-continuum elements of type C3D8R (reduced integration) and C3D8E (piezoelectric elements). As actuator voltage, the results of beam theory are applied. Note that this simulation model considers several electro-mechanical coupling effects and refinements in contrast to beam theory. Thus, this model is supposed to be suitable for a validation of the theoretical results.

The numerical results for the tip deflection $w(x = L)$ and the torsional angle $\chi(x = L)$ are shown in Fig. 15.5, for the case with and without actuation. The results show a significant reduction of the amplitude due to the actuation. The remaining vibrations are caused by the mass of the beam which has been neglected. All in all the results show a good coincidence between theoretical and Finite Element solution.

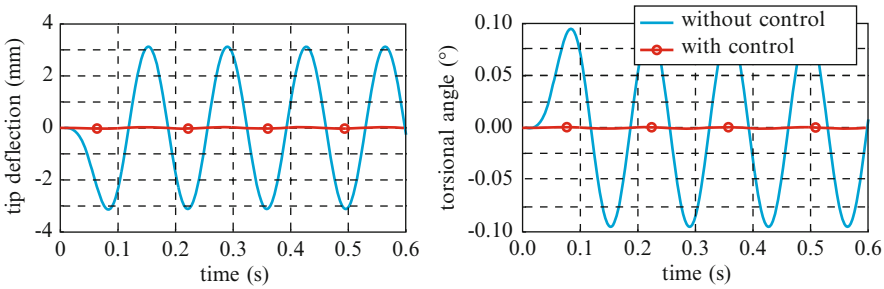



**Fig. 15.5** Numerical simulation results for the beam end $x = L$

### *15.7.2 Rotating Beam—Experimental Verification*

As a second example, a rotating beam with rectangular hollow cross-section is considered as schematically shown in Fig. 15.6. The according laboratory setup is shown in Fig. 15.7. On the beam, i.e. inside and outside of the hollow cross-section, a number of 48 piezoelectric patches has been applied. In order to reduce the number of circuits for the electrical power supply, groups of three patches have been connected. Between these groups strain gauges have been applied for sensing. For monitoring, an acceleration sensor has been placed at the free beam end.

The effective excitation of the beam is the transversal distributed inertial force per unit length

$$f_z^e = -\rho A x^2 \ddot{\varphi}, \tag{15.19}$$

caused by the rigidbody rotation angle $\varphi(t)$. Inserting into (15.16) and integrating with respect to the axial coordinate $x$ yields the actuating moment

$$M_y^a = \tfrac{1}{3}\,\rho A L^3 \ddot{\varphi}(1-\xi)^2\left(1+\tfrac{1}{2}\,\xi\right). \tag{15.20}$$

This cubic spatial distribution is discretized by means of four groups of three patches as shown in Fig. 15.8. The actuating moment of a piezoelectric patch is obtained from (15.14). With the Young modulus of the patch $E_p$, the width $b$ and the height $h$ of the beam, we obtain the actuating bending moment of the $i$-th patch

$$M_y^{a,i} = \tfrac{1}{2}\,E_p d_{31} b(h+h_L) k_i U_e l(t), \tag{15.21}$$

where the gains $k_i$ are weighting factors in order to realize the cubic distribution of the actuating moment as given in (15.20). Following the strategy presented in Ref. [16], the coefficients are found to be $k_1 = 1$, $k_2 = \frac{1}{2}$, $k_3 = \frac{1}{5}$ and $k_4 = \frac{1}{28{,}5}$.

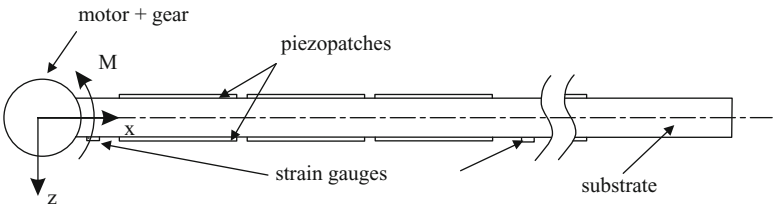



**Fig. 15.6** Rotating beam with piezoelectric patches


**Fig. 15.7** Experimental setup

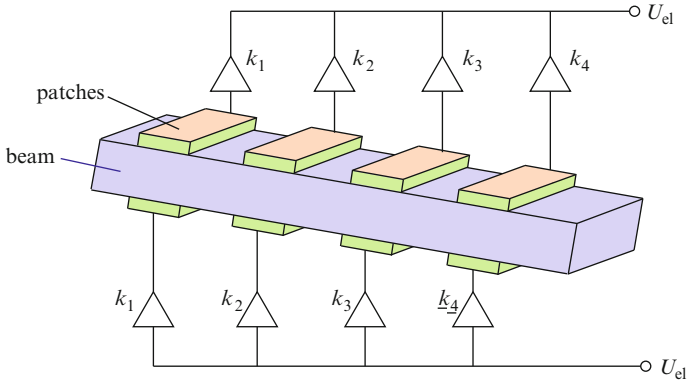



**Fig. 15.8** Patch discretisation

Figure 15.9 shows the control strategy consisting of two parts: First, the feed forward shape control $M_a^{ff}$ is implemented with an estimation of the rigidbody acceleration $\ddot{\varphi}$.

Due to several uncertainties of the system it is not possible to completely compensate the vibrations by feed forward control only. Thus, strain gauges between the patches are used as sensors to measure the average curvature $\bar{\kappa}$. The error of the curvature $e_k$, i.e. the difference of prescribed and measured curvature, is the input of the feedback controller. As a first account, a P-control law has been implemented. For monitoring, the acceleration $a_L = a(x = L)$ of the free beam end is measured.

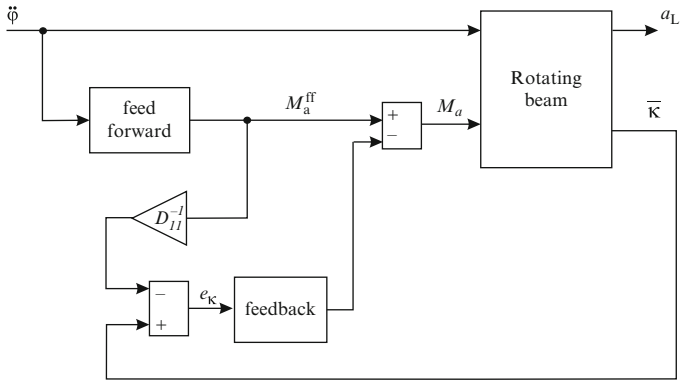



**Fig. 15.9** Control strategy

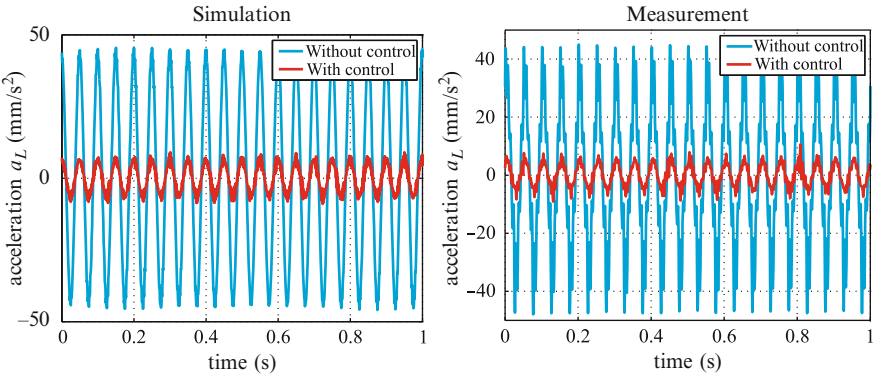



**Fig. 15.10** Time response for harmonic excitation of the first eigenfrequency

In order to optimize the control parameters, a simulation model has been implemented using the multibody dynamics simulation code HOTINT. The model is based on a Finite Element Formulation using Bernoulli-Euler beam elements which considers large deformation strains as well as the varying stiffness due to the piezoelectric patches. A motor model is implemented considering stiffness, damping and friction. The parameters of the simulation model have been calibrated to the experiment using appropriate identification strategies.

As a first investigation, the motor angle has been prescribed in sinusoidal form, the frequency coinciding with the first eigenfrequency of the beam, $f_1 = 20$ Hz. Figure 15.10 shows a comparison of simulation (left picture) and experiment (right picture) for the tip acceleration of the beam. In both cases, the amplitude of the vibration is reduced significantly. The results show a very good coincidence even in the considered resonant case.

As a second example, a triangular velocity profile has been prescribed as rigidbody motion. Figure 15.11 shows the measured time response and the

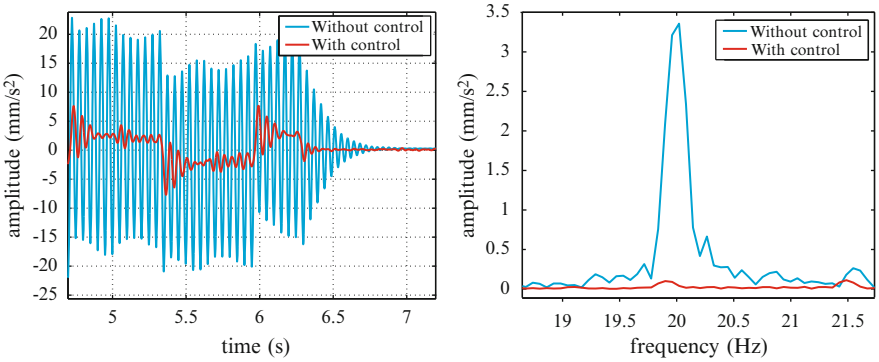



**Fig. 15.11** Time response of a triangular velocity profile

frequency response of the tip acceleration, i.e. at the free beam end. The result shows that the amplitude of the vibration is reduced significantly with the implemented control strategy.

## 15.8 Conclusions

In this paper, piezoelectric vibration control of three-dimensional flexural and torsional beam vibrations has been treated. External and inertial forces due to rigidbody motions are considered as excitations. The equations of motion have been derived in the framework of Bernoulli-Euler beam theory, and an extension of Saint Venant's theory of torsion. Laminated cross-sections and the influence of piezoelectric strains are considered. In the framework of beam theory, an exact shape control solution has been presented, i.e. the distribution of piezoelectric actuation strains in order to completely compensate the elastic beam vibrations. For the practical realisation, a patch approximation has been introduced. The theoretical results have been verified by means of numerical and experimental investigations, showing a very good coincidence. The results also show that a significant reduction of the flexible vibrations is possible with the presented strategy.

**Acknowledgements** Support of the authors from the K2-Comet Austrian Center of Competence in Mechatronics (ACCM) and G. Zenz from the Austrian Science Fund (FWF Translational project I337-N18, Dynamic response of nonlinear problems with large rotations) is gratefully acknowledged.

## Appendix

Lead zirconate titanate (PZT)

$$\boldsymbol{Q} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & 0 & 0 & 0 \\ Q_{12} & Q_{11} & Q_{13} & 0 & 0 & 0 \\ Q_{13} & Q_{13} & Q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix}, \quad \boldsymbol{d} = \begin{bmatrix} 0 & 0 & d_{31} \\ 0 & 0 & d_{32} \\ 0 & 0 & d_{33} \\ 0 & d_{24} & 0 \\ 0 & 0 & d_{15} \\ 0 & 0 & 0 \end{bmatrix}. \quad (15.22)$$

Ammonium dihydrogen phosphate (ADP)

$$\boldsymbol{Q} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & 0 & 0 & 0 \\ Q_{12} & Q_{11} & Q_{13} & 0 & 0 & 0 \\ Q_{13} & Q_{13} & Q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix}, \quad \boldsymbol{d} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ d_{25} & 0 & 0 \\ 0 & d_{25} & 0 \\ 0 & 0 & d_{36} \end{bmatrix}. \quad (15.23)$$

Macro fiber composite (MFC)

$$\boldsymbol{Q} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & 0 & 0 & Q_{16} \\ Q_{12} & Q_{11} & Q_{13} & 0 & 0 & Q_{16} \\ Q_{13} & Q_{13} & Q_{33} & 0 & 0 & Q_{36} \\ 0 & 0 & 0 & Q_{55} & Q_{45} & 0 \\ 0 & 0 & 0 & Q_{45} & Q_{55} & 0 \\ Q_{16} & Q_{16} & Q_{36} & 0 & 0 & Q_{66} \end{bmatrix}, \quad \boldsymbol{d} = \begin{bmatrix} d_{11} & d_{12} & 0 \\ d_{12} & d_{11} & 0 \\ d_{13} & d_{13} & 0 \\ 0 & 0 & d_{35} \\ 0 & 0 & d_{35} \\ d_{16} & d_{16} & 0 \end{bmatrix}. \quad (15.24)$$

## References

1. Mason WP (1981) Piezoelectricity, its history and applications. J Acoust Soc Am 6:1561–1566
2. Irschik H (2002) A review on static and dynamic shape control of structures by piezoelectric actuation. Eng Struct 24:5–11
3. Zehetner C, Irschik H (2005) Displacement compensation of beam vibrations caused by rigid-body motions. Smart Mater Struct 14:862–868
4. Zehetner C, Krommer M (2011) Control of torsional vibrations in piezolaminated rods. Struct Contr Health Monit. doi:10.1002/stc.455
5. Zehetner C, Zellhofer M, Krommer M (2011) Piezoelectric torsional sensors and actuators—a computational study. In: Papadrakakis M, Lagaros ND, Fragiadakis M (eds) Proceedings of ECCOMAS thematic conference on computational methods in structural dynamics and earthquake engineering, Corfu, Greece, 26–28 May 2011

6. Zehetner C, Gerstmayr J (2010) Compensation of flexible vibrations in a two-link robot by piezoelectric actuation. In: Irschik H, Krommer M, Watanabe K, Furukawa T (eds) Mechanics and model based control of smart materials and structures, 1st Japan-Austria joint workshop, Linz, Austria, 22–23 Sep 2008, pp 205–214
7. Zehetner C, Gerstmayr J (2010) A continuum mechanics approach for smart beams: applications. In: Topping BHV, Adam JM, Pallarés FJ, Bru R, Romero ML (eds) Proceedings of the tenth international conference on computational structures technology. Civil-Comp Press, Stirlingshire, UK, doi: 10.4203/ccp.93.215
8. Zehetner C, Zenz G, Gerstmayr J (2011) Piezoelectric control of flexible vibrations in rotating beams: an experimental study. In: Proceedings in applied mathematics and mechanics (PAMM), vol 11, pp 77–78, doi:10.1002/pamm.201110030
9. Gerstmayr J (2009) A C++ environment for the simulation of multibody dynamics systems and finite elements. In: Arczewski K, Fraczek J, Wojtyra M (eds) CD-Proceedings of ECCOMAS thematic conference: multibody dynamics, Warschau, Poland
10. Tauchert TR (1992) Piezothermoelastic behavior of a laminated plate. J Ther Stress 15:25–37
11. Mura T (1991) Micromechanics of defects in solids. Kluwer, Dordrecht
12. Irschik H, Krommer M, Belyaev AK, Schlacher K (1998) Shaping of piezoelectric sensors/actuators for vibrations of slender beams: coupled theory and inappropriate shape functions. J Intell Mater Syst Struct 9:546–554
13. Zehetner C (2008) Compensation of torsion in rods by piezoelectric actuation. Arch Appl Mech 78:921–933
14. Rand O, Rovenski V (2005) Analytical methods in anisotropic elasticity. Birkhäuser, Boston
15. Zehetner C (2005) Piezoelectric compensation of flexural and torsional vibrations in beams performing rigid-body motions. Doctoral thesis, Schriften der Johannes Kepler Universität Linz, Trauner Verlag, Linz
16. Nader M, Kaltenbacher M, Krommer M, von Garssen HG, Lerch R (2006) Active vibration control of a slender cantilever using distributed piezoelectric patches. In: Proceedings of the thirteenth international congress on sound and vibration, Vienna, Austria

# Chapter 16
# Multibody Dynamics Approaches to Biomechanical Applications to Human Motion Tasks

**Jorge A.C. Ambrosio**

**Abstract** Applications of multibody dynamics or control to human mobility, impact biomechanics, ergonomics or health and medical cases require that reliable models of human body, including all relevant anatomical segments and a representation of the musculoskeletal system, are developed. The system state variables are available either to a control algorithm or to appraise the internal forces or even to evaluate performance indexes associated to the particular task. Here, a biomechanical model of the human body is presented and applied to demonstrate the basic modeling requirements. A strategy for the control of the biomechanical model motion, based on a distributed hierarchical control, is proposed. The biomechanical model is used to study zero momentum maneuvers, such as those of an astronaut in space or of a high-platform diver. Recognizing that the internal driving forces in the human body result from the musculoskeletal system and not from torque actuators, a procedure to evaluate the muscle forces is presented. Muscle activation dynamics models and optimization techniques are part of the proposed methodology. A human locomotion task demonstrate the procedure and to show the relation between muscle forces and the joint torques used in the control model.

## 16.1 Introduction

Biomechanical models of different animals may be required to plan their motion either for coordination studies, training practices, ergonomics, health applications or others. The use of joint torque actuators, as in robotic applications, may simplify the dynamic description of the problem but presents difficulties in terms of the significance of their relation to muscle forces. For animals with exoskeleton, such as crabs, the musculoskeletal system is composed basically of pairs of muscles, for

J.A.C. Ambrosio (✉)
Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
e-mail: jorge@dem.ist.utl.pt

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_16, 

each joint, that act as push-rods and for which the relations between joint torques and muscle forces is unique [1, 2]. For animals with an internal skeleton, such as the mammalians, the musculoskeletal system is complex and highly redundant [2]. In this case the relations between muscle forces and joint torques are not obvious. When considering motion coordination activities of humans the independent control of the joint torques is already a complex task, in the limit of the feasibility, being the independent control of the muscle forces with the objective of planning general human motions unfeasible. Therefore, a procedure to break the problem of human motion planning into manageable problems is of upmost importance.

The first step is the construction of a biomechanical model of the human body that can be used in the coordination studies foreseen in this work. The biomechanical models applied on the study of the human locomotion require that the major anatomical segments of the human body are represented by rigid bodies. The biomechanical model used is composed of 16 anatomical segments represented by 16 rigid bodies with their physical characteristics obtained from a human anatomic database to ensure its biofidelity [3]. The anatomical joints are represented either by kinematic joints or by contact joints in the multibody model, depending on the objectives of the analysis. The ligaments and other passive tissues required to provide stability or stiffness to the anatomical joints are typically represented as spring-damper elements with linear or nonlinear characteristics. In the biomechanical model used here the anatomical joints are modeled as mechanical joints and the ligaments are substituted by a torque penalization of any relative motion that is excessive [3].

A methodology suitable to the control large scale system is necessary to allow for planning general purpose motion of the human body [4]. A decentralized and hierarchical control is used here to implement a feasible controller [5]. This model has three stages of controllers of increasing complexity: low-level controllers to act on the individual joints; intermediate controllers to coordinate the prescribed motion of each limb; a high-level controllers to select the motion of the limbs and the period of time in which they have to take place. The execution of a large number of maneuvers is then planed using optimal decision approaches [6]. In this work, a strategy for the angular reorientation of multibody systems using zero momentum turns is presented to demonstrate the decentralized and hierarchical control implemented. Space satellites with movable antennas, astronauts, divers or the cat self-aligning reflex are examples of these types of systems [5, 7–9].

The biomechanical model is driven through the motion defined by the control strategy, or acquired experimentally, by joint actuators that drive the degrees-of-freedom of the biomechanical model associated with joints, or by muscle actuators that drive the degrees-of-freedom of the joints crossed by the muscles. Being the motion of the biomechanical model the same, there is equivalence between the joint torques and the muscle forces. The use of optimization procedures that allow for the calculation of the redundant muscle forces, generated in a particular muscle apparatus of the human body is the key methodology to obtain such equivalence. The problem of finding the internal forces of the biomechanical system that lead to a prescribed motion can be defined as an optimal problem where the objective is to find suitable joint net-moments-of force or muscle forces [10–13]. The problem can

be solved by an inverse dynamic analysis where the objective is simply to minimize the physiological or metabolic criteria, being the motion and contact forces with the external environment the data required for the optimal problem. Here a Hill type muscle model is applied, being the force produced by the muscle contractile element calculated as a function of the muscle activation, maximum isometric peak force, muscle length and muscle rate of shortening. The equations of motion of the biomechanical system and the performance criteria used in de optimization procedure are expressed in terms of muscle activations instead of muscle forces [14, 15]. The procedure is exemplified through the application of the methodology described in this work to a motion acquired in the Lisbon Biomechanics Laboratory.

## 16.2 Overview of the Multibody Formulation

A multibody system is defined as a collection of rigid and/or flexible bodies constrained by kinematic joints and eventually acted upon by a set of internal and/or external forces. Different sets of coordinates can be used to describe the multibody system such as those presented in Fig. 16.1. Using Cartesian coordinates, shown in Fig. 16.1a, the position and orientation of each body $i$ in the space is described by a position vector $\mathbf{r}_i$ and a set of rotational coordinates $\mathbf{p}_i$ organized in a vector as [16]

$$\mathbf{q}_i = [\mathbf{r}^T, \mathbf{p}^T]_i^T \tag{16.1}$$

A multibody system with $nb$ bodies is described by a set of coordinates as

$$\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T, \cdots, \mathbf{q}_{nb}^T]^T \tag{16.2}$$

The dependencies among system coordinates, which result from the existence of mechanical joints interconnecting the several bodies, are defined through the introduction of kinematic relationships written as

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \tag{16.3}$$

where $t$ is the time variable, which is used only for the driving constraints. The second time derivative of (16.3) with respect to time yields

$$\ddot{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{0} \quad \equiv \quad \mathbf{\Phi}_\mathbf{q}\, \ddot{\mathbf{q}} = \boldsymbol{\gamma} \tag{16.4}$$

where $\mathbf{\Phi}_\mathbf{q}$ is the Jacobian matrix of the constraints, $\ddot{\mathbf{q}}$ is the acceleration vector and $\boldsymbol{\gamma}$ is the vector that depends on the velocities and time.

The equations of motion for a constrained multibody system (MBS) of rigid bodies, such as the biomechanical model, are written as

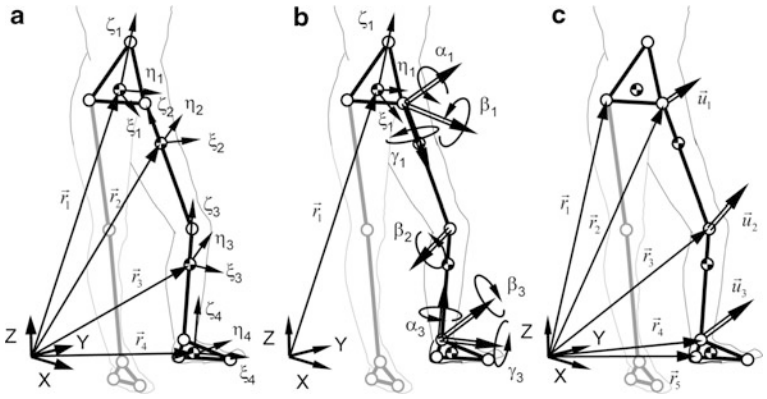



**Fig. 16.1** Types of coordinates: (**a**) Cartesian coordinates, (**b**) joint coordinates, (**c**) natural coordinates

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{g} + \mathbf{g}^{(\mathbf{c})} \tag{16.5}$$

where $\mathbf{M}$ is the system mass matrix, $\ddot{\mathbf{q}}$ is the vector that contains the state accelerations, $\mathbf{g}$ is the generalized force vector, which contains all external forces and moments, and $\mathbf{g}^{(c)}$ is the vector of constraint reaction equations. The joint reaction forces can be expressed in terms of the Jacobian matrix of the constraint equations and the vector of Lagrange multipliers

$$\mathbf{g}^{(c)} = -\boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda} \tag{16.6}$$

where $\boldsymbol{\lambda}$ is the vector that contains $m$ unknown Lagrange multipliers associated with $m$ holonomic constraints. Substitution of (16.6) in (16.5) yields

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda} = \mathbf{g} \tag{16.7}$$

In dynamic analysis, a unique solution is obtained when the constraint equations are considered simultaneously with the differential equations of motion with proper set of initial conditions. Therefore, (16.4) is appended to (16.7), yielding a system of differential algebraic equations that are solved for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. This system is given by

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{r} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{16.8}$$

The solution of the multibody equations of motion and their integration in time is depicted in Fig. 16.2. The set of differential algebraic equations of motion, (16.8) does not use explicitly the position and velocity equations associated to the kinematic constraints, (16.3) and time derivative, respectively. Thus, in order to

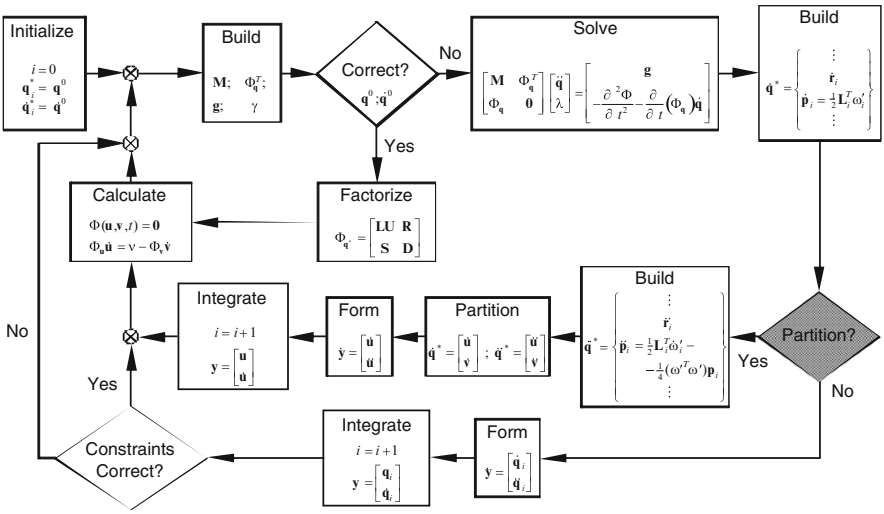



**Fig. 16.2** Flowchart with the forward dynamic analysis of a multibody system

stabilize or keep under control the constraints violation, (16.8) is solved by using the Baumgarte Stabilization Method or the augmented Lagrangean formulation, and eventually complemented by using the coordinate partition method, and the integration process is performed using a predictor–corrector algorithm with variable step and order.

## 16.3 Biomechanical Model of the Skeletal System

A biomechanical model suitable for human motion analysis requires that the different anatomical segments and their relative mobility are described, the muscle activation and corresponding forces are represented and that the skeletal-muscle apparatus is included in the model. In what follows a description of each part of the biomechanical model is provided.

The biomechanical model of the human body is defined using 16 anatomical segments and their corresponding rigid bodies is presented in Table 16.1 and illustrated in Fig. 16.3 [3]. Considering this kinematic structure, an open loop topology can be identified, with a base body described by rigid body number 1, and five kinematic branches defined by the four limbs and the head/neck. The model has 44 degrees-of-freedom that correspond to 38 rotations about 26 revolute joints and 6 universal joints, plus 6 degrees-of-freedom that are associated with the free body rotations and translations of the base body. The model presented is only one of many that can be used to the biomechanical analysis of different human motion tasks. Due to the kinematic structure, where no spherical joints is used it can be applied in inverse dynamic analysis. However, its application to a particular individual requires that its anatomical segments are properly scaled. In the present

**Table 16.1** Physical characteristics of anatomical segments and rigid bodies for the 50th-percentile human male

| Description | Body | Length | CM location | | Mass | Moments of inertia |
|---|---|---|---|---|---|---|
| | $i$ | $L_i$(m) | $d_i$(m) | $\underline{dd}_i$(m) | $m_i$(kg) | $(I_{xx}/I_{yy}/I_{zz})_i(10^{-2}\times)$ |
| Lower torso | 1 | 0.275 | 0.064 | 0.094 | 14.200 | 26.220/13.450/26.220 |
| Upper torso | 2 | 0.294 | 0.101 | 0.161 | 24.950 | 24.640/37.190/19.210 |
| Head | 3 | 0.128 | 0.020 | 0.051 | 4.241 | 2.453/2.249/2.034 |
| R upper arm | 4 | 0.295 | 0.153 | – | 1.992 | 1.492/1.356/0.248 |
| R lower arm | 5 | 0.250 | 0.123 | – | 1.402 | 1.240/0.964/0.298 |
| Hand | 13 | 0.185 | 0.093 | 0.045 | 0.489 | 0.067/0.146/0.148 |
| L upper arm | 6 | 0.295 | 0.153 | – | 1.992 | 1.492/1.356/0.248 |
| L lower arm | 7 | 0.376 | 0.180 | – | 1.892 | 1.240/0.964/0.298 |
| Hand | 14 | 0.185 | 0.093 | 0.045 | 0.489 | 0.067/0.146/0.148 |
| R upper leg | 8 | 0.434 | 0.215 | – | 9.843 | 1.435/15.940/9.867 |
| R lower leg | 9 | 0.439 | 0.151 | – | 3.626 | 1.086/3.830/3.140 |
| Foot | 15 | 0.069 | 0.271 | 0.035 | 1.182 | 0.129/0.128/2.569 |
| L upper leg | 10 | 0.434 | 0.215 | – | 9.843 | 1.435/15.940/9.867 |
| L lower leg | 11 | 0.439 | 0.151 | – | 3.626 | 1.086/3.830/3.140 |
| Foot | 16 | 0.069 | 0.271 | 0.035 | 1.182 | 0.129/0.128/2.569 |
| Neck | 12 | 0.122 | 0.061 | – | 1.061 | 0.268/0.215/0.215 |

The dimensions and positions of the center of mass locations, with respect to the proximal joint with reference to Fig. 16.3.

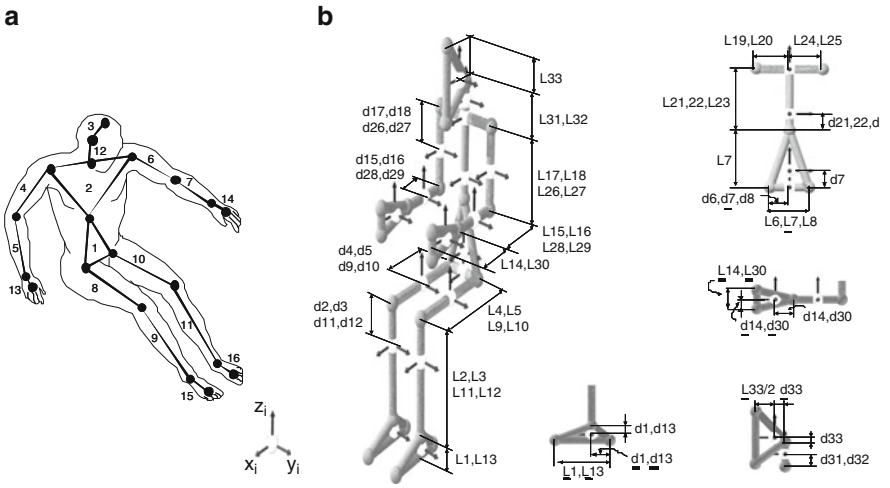



**Fig. 16.3** Biomechanical model with 16 anatomical segments: (**a**) model topology, (**b**) length and center of mass of each anatomical segment

work, the scaling procedure used calculates for each anatomical segment, non-dimensional scaling factors, based on measured data from the subject and equivalent data from the 50th percentile human male. These scaling factors are defined as [17]:

$$\chi_{L_i} = \frac{L_i^{n^{th}}}{L_i^{50^{th}}}; \quad \chi_{m_i} = \frac{m_i^{n^{th}}}{m_i^{50^{th}}}; \quad \chi_{I_i} = \chi_{m_i} \cdot \chi_{L_i}^2 \tag{16.9}$$

where $\chi_{L_i}$, $\chi_{m_i}$ and $\chi_{I_i}$ are respectively the scaling factors of the length, mass and moments of inertia calculated for segment $i$. It should be noted that if the length and mass of each segment of the subject are not available, the calculation of $\chi_{L_i}$ and $\chi_{m_i}$ can be performed using the ratio between heights and the ratio between total body weights, respectively. The length scaling factor is used to scale all the dimensions, including the location of each center-of-mass. This procedure should only be used to scale subjects of the same gender and with anthropometric characteristics not far from the reference model.

## 16.4 Task Targeted Control of the Biomechanical System

The motion of the basic biomechanical model of the human body, presented in Fig. 16.3 and Table 16.1, can be driven by torques applied at the anatomical joints. Although the human body has a complex musculoskeletal system to develop the internal forces that lead to its motion it is accepted that a torque equivalent exists for each anatomical joint at any instant [18]. Due to the high complexity of the multibody biomechanical system the use of global control through full state feedback is not recommended [4]. A decentralized and hierarchical control scheme is proposed here based on the work presented in Refs. [5, 19]. Three levels of control are defined: based on an optimal strategy, the high level controller chooses the best set of maneuvers to achieve a given result; the intermediate controllers ensure the coordination of the local controllers to ensure that the set of maneuvers defined by the high-level controller are achieved in the proper sequence; the low level controllers actuate each joint using a variable gain PD controller, as described in Fig. 16.4. Note that in each control level other control strategies can be used instead of the ones proposed here. However, the hierarchical structure of the control remains unchanged.

### *16.4.1 Low-Level Control*

The low level control is applied to each joint of the biomechanical model with the objective that its joint angle follows a prescribed time history. The kinematic topology of the biomechanical system is represented by a base body connected to other rigid bodies by kinematic joints, forming five open-loop chains. The base body, i.e. the lower torso, has six degrees-of-freedom (dof) and connects to other bodies in the chain with revolute joints, each of them with one dof, or with universal joints, which have two dof each.

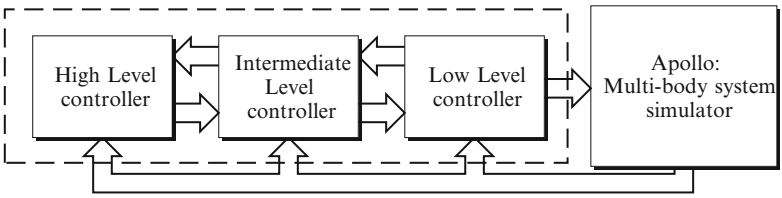



**Fig. 16.4** Block diagram of the control structure

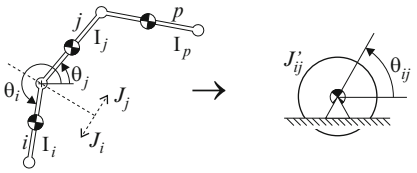



**Fig. 16.5** Simplified model of the relative rotation between two bodies, in which the doted arrows indicate the bodies use in the evaluation of $J_i$ and $J_j$

Each degree-of-freedom is controlled by an independent variable gain PD controller [20]. Each pair of bodies is connected by a joint actuated by a single independent control torque $T_{ij}$. A simplified representation of the dynamic equations of the one dof joint, disregarding Coriolis and centrifugal accelerations and joint reaction forces, is

$$J'_{ij}\,\ddot{\theta}_{ij} = T_{ij} \tag{16.10}$$

where

$$J'_{ij} = \frac{J_i J_j}{J_i + J_j} \tag{16.11}$$

$$\theta_{ij} = \theta_j - \theta_i \tag{16.12}$$

where $\theta_{ij}$ is the relative angle between adjacent body segments and $J_i$ is the inertia moment of body $i$ and of all bodies that follow in the kinematic chain, as seen in Fig. 16.5.

The control law applied is proportional-derivative (PD) on angular position:

$$T_{ij} = -k^P_{ij}\left(\theta_{ij} - \theta_{ref\ ij}\right) - k^D_{ij}\,\dot{\theta}_{ij} \tag{16.13}$$

where $k^P$ and $k^D$ are constant gain constants for the proportional and derivative parts of the controller, respectively, and $\theta_{ref\ ij}$ is the desired angular position on the joint $ij$ imposed by the intermediate controller.

The simplified model of a system with feedback is described by the second order linear differential equation given by

$$J'_{ij}\,\ddot{\theta}_{ij} + k_{ij}^{D}\,\dot{\theta}_{ij} + k_{ij}^{P}\,\theta_{ij} = k_{ij}^{P}\,\theta_{ref\,ij} \tag{16.14}$$

A more common form of presenting the equilibrium equation for a second order system is

$$\ddot{\theta}_i + 2\zeta_i\,\omega_{ni}\,\dot{\theta}_i + \omega_{ni}^2\,\theta_i = 0 \tag{16.15}$$

in which the damping coefficient $\zeta_{ni}$ and natural frequency $\omega_{ni}$ are written as

$$\zeta_i = \frac{k_{ij}^{D}}{2\omega_{ni}J'_{ij}} \quad ; \quad \omega_{ni}^2 = \frac{k_{ij}^{P}}{J'_{ij}} \tag{16.16}$$

Equation 16.16 is the key to define the controller gains. The choice for $k_{ij}^{D}$ is done such a way that the system becomes slightly over damped. Rearranging the terms in (16.16) and substituting them in (16.13) results in the control law:

$$T_{ij} = -J'_{ij}\,\omega_{ij}^2\left(\theta_{ij} - \theta_{ref\,ij}\right) - \alpha\sqrt{2}\,J'_{ij}\omega_{ij}\dot{\theta}_{ij}, \qquad \alpha \geq 1 \tag{16.17}$$

where $\omega_{ij}$ is the angular frequency of the controller and $\alpha$ is the over-damping term. Note that the angular frequency of the controller, which needs to be adjusted for each particular joint, defines how stable and fast is the response of the system with feedback.

Due to the kinematic chain of the multibody system, the equivalent moment of inertia varies if any of the bodies that precede the current joint changes its center of mass. Referring to Fig. 16.5, the update of the moments of inertia, in each time step, is

$$J_i = \mathbf{u}_{ij}^T\left\{\sum_{k\in S_i}\mathbf{A}_k^T\,\mathbf{I}_k\,\mathbf{A}_k\right\}\mathbf{u}_{ij} + \sum_{k\in S_i} m_k\,d_{ij,k}^2 \tag{16.18}$$

where $\mathbf{u}_{ij}$ is a unitary vector collinear to the axis of the joint $ij$, $S_i$ is the set of bodies that precede to body $i$ in the kinematic chain, $\mathbf{A}_k$ is the coordinate transformation matrix from the local body referential to the global referential and $d_{ijk}$ is the distance between the axis of joint $ij$ and the center of mass of body $k$ given by

$$d_{ij,k} = \left\| \mathbf{r}_{ij,k} - \left(\mathbf{r}_{ij,k}^T\mathbf{u}_{ij}\right)\mathbf{u}_{ij}\right\| \tag{16.19}$$

where $\mathbf{r}_{ij,k} = \mathbf{r}_k - \mathbf{r}_{ij}$ is the spatial position of joint $ij$. Note that the update of the equivalent inertia moments, for each configuration of the kinematic chain, constitutes in fact a variation in the control gains.

In order to avoid that the controllers develop an unrealistic moment a saturation moment is adopted for each controller, defined as

$$F_{ij}(T_{ij}) = \begin{cases} -T_{ij\,\max} & if \quad T_{ij} < -T_{ij\,\max} \\ T_{ij} & if \quad |T_{ij}| \leq T_{ij\,\max} \\ T_{ij\,\max} & if \quad T_{ij} > T_{ij\,\max} \end{cases} \tag{16.20}$$

The thresholds used for the control saturation can be based on physiological reasoning. It must be emphasized that such threshold may vary by large amounts for different individuals.

### 16.4.2 Intermediate-Level Control

The intermediate control, after receiving from the higher level control the definition of the maneuver that the whole body is supposed to develop and the time period in which it has to be developed, defines for each local controlled the reference angle and a law for its variation in time. The type of maneuvers that are supervised by the intermediate controller are exemplified in Fig. 16.6. Each of the maneuvers, such as opening and closing arms, is designed and stored in a library of maneuvers that is made available to the high-level controller.

Each of the maneuvers that an intermediate controller supervises has costs, which are specified here in terms of energy consumption. The energy function associated to a particular maneuver is the sum of the energy consumed by all the low-level controllers, under the supervision of the intermediate-level controller, to perform the rotations required for the particular maneuver. A simplified measure of the energy necessary for a PD low-level controller to develop a prescribed rotation by recognizing that (16.14) has an analytical solution of the type

$$\theta(t) = \theta_{ref} + C_1 e^{\frac{-k^D - \sqrt{(k^D)^2 - 4Jk^P}}{2J}t} + C_2 e^{\frac{-k^D + \sqrt{(k^D)^2 - 4Jk^P}}{2J}t} \tag{16.21}$$

Assuming that the initial conditions are $\theta(0) = \dot{\theta}(0) = 0$ the solution of (16.21) for the constants $C_1$ and $C_2$ is

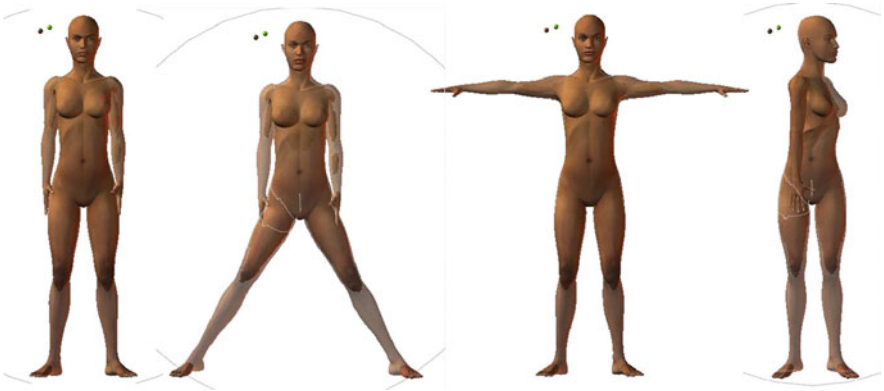


**Fig. 16.6** Typical maneuvers handled by the intermediate controllers, from left to right: (1) initial posture; (2) legs opening/closing; (3) arms opening/closing; (4) upper body clockwise/counter-clockwise rotation

$$\begin{aligned} C_1 &= -\frac{-k^D+\sqrt{\left(k^D\right)^2-4Jk^P}}{2\sqrt{\left(k^D\right)^2-4Jk^P}}\theta_{ref} = \alpha_1\theta_{ref} \\ C_2 &= -\frac{k^D+\sqrt{\left(k^D\right)^2-4Jk^P}}{2\sqrt{\left(k^D\right)^2-4Jk^P}}\theta_{ref} = \alpha_2\theta_{ref} \end{aligned} \tag{16.22}$$

Substituting (16.22) in (16.21) results in

$$\theta(t)=\theta_{ref}\left(1+\alpha_1 e^{\frac{-k^D-\sqrt{\left(k^D\right)^2-4Jk^P}}{2J}t}+\alpha_2 e^{\frac{-k^D+\sqrt{\left(k^D\right)^2-4Jk^P}}{2J}t}\right)=\theta_{ref}\,\varphi(t) \tag{16.23}$$

upon substitution of (16.23) into (16.13) results in the equation for the torque applied in the joint by the low-level controller as

$$T(t) = -k^P\theta_{ref}(\varphi(t)-1) - k^D\theta_{ref}\dot{\varphi}(t) = \theta_{ref}\bar{T}(t) \tag{16.24}$$

A measure of the energy used by the controller in a given time interval $t_f$ is

$$\int_0^{t_f} T(t)\theta(t)dt = \theta_{ref}^2\int_0^{t_f}\bar{T}(t)\,\varphi(t)dt = \theta_{ref}^2 E(t_f) \tag{16.25}$$

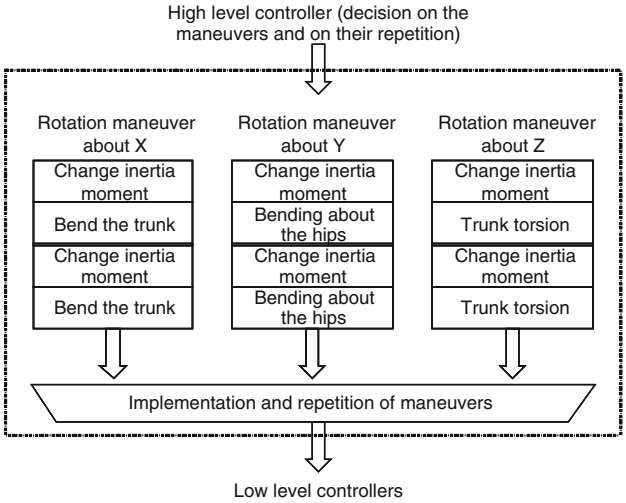



**Fig. 16.7** Schematic description of the high-level controller planning of the sequences of maneuvers to be supervised by the intermediate level controllers

Assuming a constant gain for the PD controllers and a constant activation time (16.25) shows that the energy used in each low-level controller is proportional to the square of the reference angle that the intermediate-level controller associates to its action. Therefore, the energy required for each complete maneuver of an intermediate-level controller is obtained as the sum of all energies used in the low-level controllers under its supervision.

### *16.4.3 High-Level Control*

Given an objective for the human motion task the high-level controller has to plan the different maneuvers that the intermediate-level controllers have to supervise. A library of maneuvers, such as arm stretching, legs opening, arms folding, legs stretching, body torsion clockwise or any other, provides the high-level controller the choice of basic maneuvers that can be used to compose the more complex motion. For instance, for zero momentum turns of the human body about the vertical, side or front axis the sequence of maneuvers depicted in Fig. 16.7 are devised.

The collection of maneuvers available to the high-level controller may be large having different sequences to achieve the same result. The high-level controller selects the sequence that minimizes a cost function defined as.

$$\begin{aligned} & f_E(i,\theta) = d_i|\theta| + c_i(1 + floor(|\theta|/\theta_{\max i})), \quad i \in \{X,Y,Z\} \\ & floor(x) = n: \quad n \le x \wedge x - n < 1, \quad n \in IN \end{aligned} \tag{16.26}$$

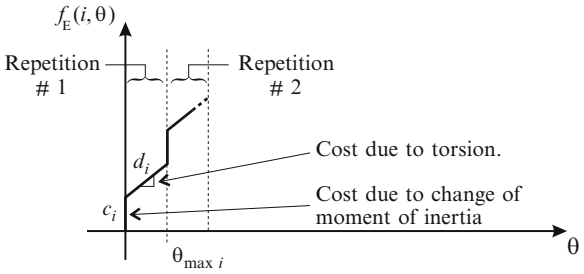



**Fig. 16.8** Energy cost function associated to a particular zero angular momentum turns motion for which a repetition is necessary

where $c_i$ represent the cost due to the positioning of the body segments such a way that the next maneuver may be performed, and $d_i|\theta|$ represents the cost of developing the next maneuver in line. The graph in Fig. 16.8 shows how the energy requirements increase with the complexity of the planned maneuvers. It should be noticed that often it is necessary that several repetitions of the sequence of maneuvers must be done to achieve the final objective. Notice also that the intermediate controller simply guarantees that the maneuver assigned by the high level controller is fulfilled, being its cost known beforehand.

The design of the library of the basic maneuvers, to be supervised by the intermediate controllers, may be more or less elaborate. Each one of the maneuvers may be optimal, by given criteria, or simply designed to achieve an end result. The optimal design of these maneuvers constitutes a topic in itself and, therefore, is not the subject of this work. In what follows it is assumed that a collection of maneuvers, such as those depicted in Fig. 16.6, are available for the high-level controller to select.

## 16.5 Application to the Human Body Attitude in Zero Gravity

The self-aligning reflex of the cat, the alignment of a high-platform diver or the pirouettes of a skater, are some of the best known maneuvers that involve zero-momentum turns [5, 7, 9].

The sequence described in Fig. 16.9a illustrates the maneuvers involved in the self-alignment reflex of the cat, characterized by successions of torsion-bending tasks while in Fig. 16.9b describes torsion tasks with changes in the inertia moments, associated with human body alignments.

For the human body self-alignment zero momentum turns the change in the inertia moments of the upper and lower parts of the body are achieved by stretching and closing arms or legs. For instance, the realignment of the body in the sagittal plane, i.e., about the Y axis, is achieved with the maneuvers depicted in Fig. 16.10 while those shown in Fig. 16.11 lead to a realignment about the Z axis.

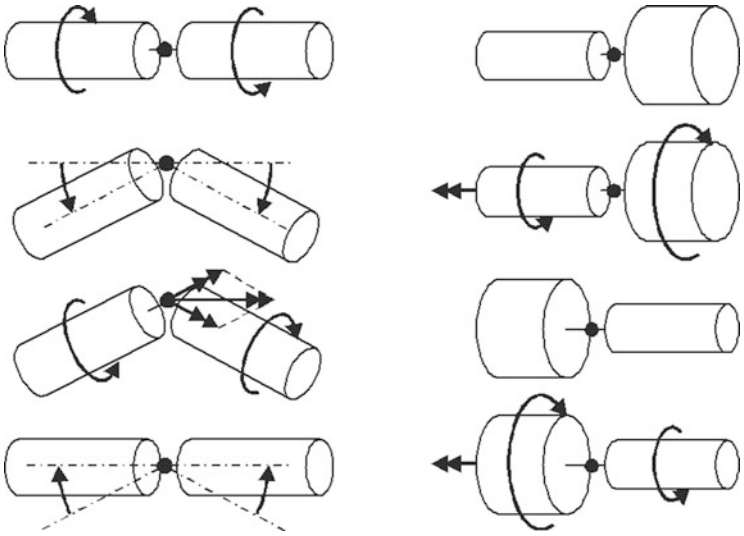


**Fig. 16.9** Maneuvers that involve zero-momentum turns: (*left*) torsion-bending as in the cat self-aligning reflex; (*right*) torsion with variation of the inertia moments for human re-orientation tasks

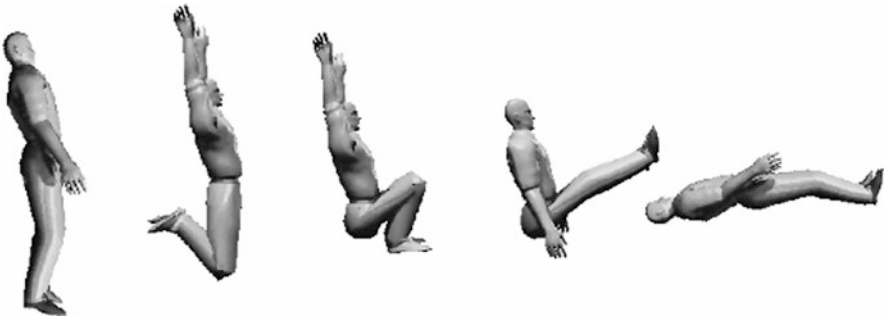


**Fig. 16.10** Zero angular momentum maneuvers illustrating a rotation in Y axis

**Fig. 16.11** Zero angular momentum maneuvers illustrating a rotation in Z axis

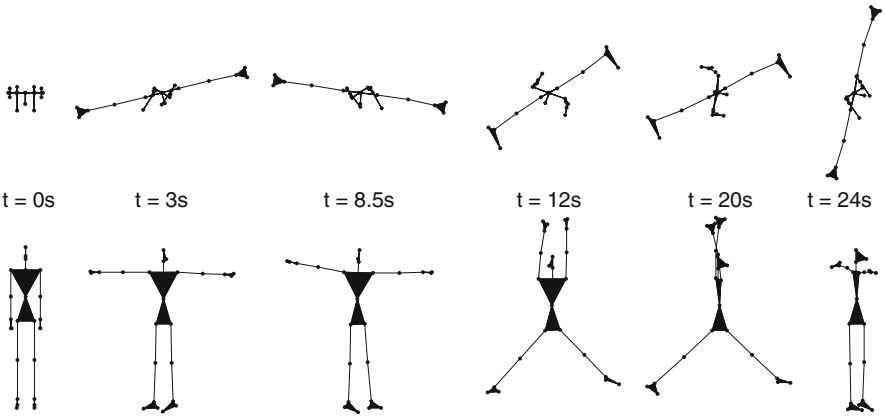



**Fig. 16.12** Initial, intermediate and final position of biomechanical model: *top view*; *front view*

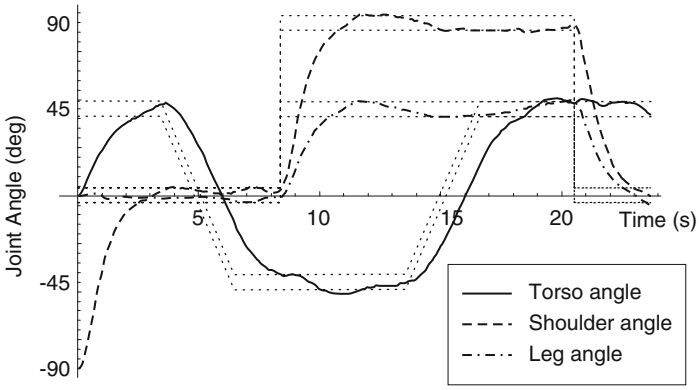



**Fig. 16.13** Change of angles during maneuvers for shoulder, torso and leg. The *dotted lines* represent the limits of the angles within which there is no active control

As an application of the methodology the spatial change of attitude of a biomechanical model in a zero gravity environment is selected. A model initially facing the X-axis is required to rotate 60° counter clockwise, resulting in the motion illustrated by Fig. 16.12. The general purpose multibody code APOLLO is used [21]. In what follows no particular effort is put in using optimized gains for the low-level controllers. In setting up the gains the physiological characteristics of the human body model must be accounted for [22].

In Fig. 16.13 the change of the angles due to the control strategy adopted are represented as a function of time. The corridors shown in the picture correspond to the range for which a particular low level controller is not active because the corresponding joint angle is in the neighborhood of the target.

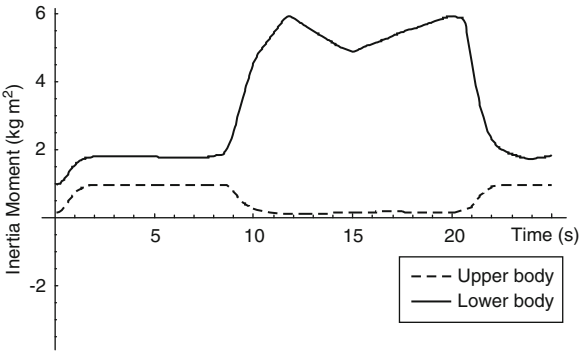



**Fig. 16.14** Variation of the upper body and lower body inertia moment about the Z axis

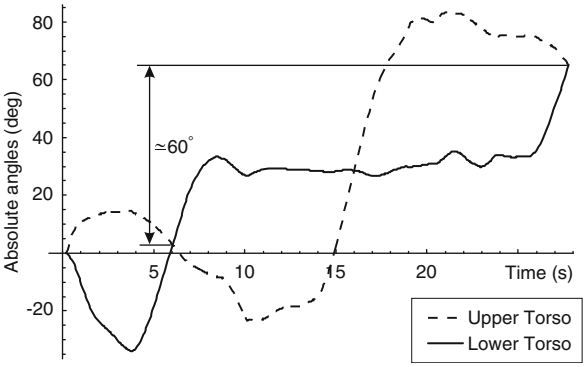



**Fig. 16.15** Evolution of the attitude upper and lower torso during the simulation

The evolution of the total moments of inertia about Z-axis for upper torso and the upper extremities of the biomechanical model and for the lower torso and lower extremities, respectively, are presented in Fig. 16.14. It is observed that the variation of the inertia moment of the lower torso and legs is more important that the variation of the upper torso inertia. As a consequence, the rotation of the lower torso is much smaller than the upper torso while the control is trying to reposition the biomechanical model.

The orientation of the body in the alignment task is depicted in Fig. 16.15. It is observed that the upper and lower torso start with an equal orientation of 0°, i.e., the biomechanical model is looking along the X axis and both finish with an equal orientation of 60°, reaching the goal set for this motion task.

It should be noted that the individual maneuvers supervised by the intermediate-level controllers are selected among the few maneuvers designed and made available to the high-order controller, none of them being optimal. The collection of maneuvers implemented, at this time, only allows for zero-momentum turns about

the principal axis and not about an arbitrary axis. Nevertheless, it the implementation of other human motion tasks can be done following the same procedure defined here.

Another aspect of the procedure described concerns the relations between the gains and the physiologic characteristics of the models. Higher gains for the low-level controllers lead to a higher speed at which a task is developed but may also result in unrealistic torques that are not possible with the musculoskeletal system. Therefore an appraisal of the relation between torques at the anatomical joints and forces in the muscles must be devised. Here, the relation between these torques and the muscle forces is done for the locomotion system in a gait task. The same methods used hereafter can be applied to obtain the muscle forces necessary to develop other human motion tasks and evaluate the feasibility of the torques required by the low-level controllers.

## 16.6 Inverse Dynamic Analysis of the Biomechanical System

The inverse dynamic analysis of the biomechanical system consists in finding the internal forces that lead to a known motion of the system. The kinematics of the human subject, who is represented by the biomechanical model, is obtained either by using a set of cameras and motion reconstruction techniques or by designing it, such as the decentralized hierarchical control does. When the muscle forces are represented by their joint torque equivalents, or net moments-of-force, the number of unknown torques is equal to the number of equations of motion and, for each time frame, the inverse dynamic problem is determined.

However, the human muscle system is highly redundant, being possible to identify several muscles that guide the same degree-of-freedom of the same anatomical joint. In this case, for each time frame, there are many solutions for the muscle force sharing and the inverse dynamic problem is not determined. The solution of the muscle force sharing problem requires the use of optimization techniques to relate the muscle activity with a merit function that represents the objective of the motion. Therefore, for a given motion there is a unique set of joint torques that causes it. By using an optimization approach, for the given motion there is a distribution of muscle forces that best fits the kinematics of the biomechanical model. Consequently, the muscle forces evaluated in this form and the joint torques are uniquely related. This relation between joint torques and muscle forces allows appraising the feasibility of the control torques necessary to achieve a given maneuver.

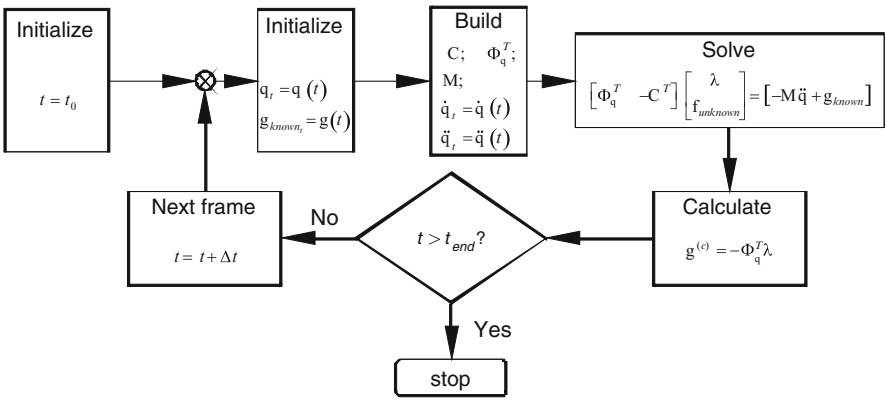



**Fig. 16.16** Schematic description of the procedure used in the solution of the inverse dynamics problem

## *16.6.1 The Deterministic Problem: Joint Torques*

### 16.6.1.1 Inverse Dynamic Analysis

The inverse dynamics problem can be setup using the multibody equation of motion, described by (16.7). Let it be assumed that the prescribed motion of the model is fully known and consistent with the kinematic constraints of the biomechanical model. Let the force vector be partitioned into a vectors of know forces $\mathbf{g}_{known}$ and unknown forces $\mathbf{g}_{unknown}$. In (16.7) the only unknowns are the Lagrange multipliers $\boldsymbol{\lambda}$ and the unknown applied forces $\mathbf{f}_{unknown}$. Equation 16.20 is rearranged as

$$\left[\boldsymbol{\Phi}_{\mathbf{q}}^T - \mathbf{C}^T\right] \begin{Bmatrix} \boldsymbol{\lambda} \\ \mathbf{f}_{unknown} \end{Bmatrix} = \{-\mathbf{M}\ddot{\mathbf{q}} + \mathbf{g}_{known}\} \tag{16.27}$$

where matrix $\mathbf{C}$ is used to map the space of the forces into the space of coordinates that describes the system being its structure dependent on the type of applied force. The solution of (16.27), highlighted in Fig. 16.16, is obtained for a finite number of time instants which depends on the sampling of the system kinematics. The solution obtained for a particular time instant is fully independent of that in any other instant.

The solution of the linear system of (16.27) is unique if the number of independent kinematic constraints and unknown forces is equal to the number of coordinates of the biomechanical system. Otherwise, the solution is not unique due to the redundant set of forces and/or constraints used and the solution of the problem has to be obtained by defining suitable criteria and using optimization methodologies. The use of this type of approaches is exemplified by Silva and Ambrósio [23].

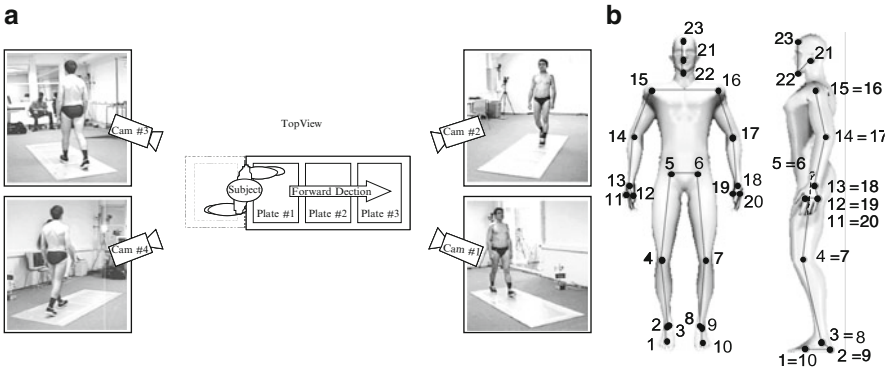



**Fig. 16.17** Kinematic and kinetic data acquisition in an experimental environment: (**a**) Experimental setup; (**b**) Anatomical landmarks

### 16.6.1.2 Kinematic and Kinetic Data Acquisition

The data required for the inverse dynamic analysis of the biomechanical model has to be designed, as the controller does, or acquired experimentally. The setup for the data acquisition process is depicted in Fig. 16.17 and it is used here to collect the data that is used to show the relation between the joint torques and the muscle forces.

For a case of gait analysis, used here to exemplify the procedure, the trial starts at the time step just before right heel contact with the floor, and continues until the subsequent occurrence of the same event in the same foot. During the stride period, the subject walks over three force plates that measure the ground reaction forces for both feet [23, 24]. The gait analysis of a 25-year-old male subject with a height of 1.70 m and a total body mass of 70 kg wearing running shoes is used here. A total number of 66 frames are recorded with a sampling frequency of 60 Hz. The trial has a total duration of 1.083 s that corresponds to a walking cadence of approximately 111 steps per minute, which is a normal cadence stride [22].

### 16.6.1.3 Results of the Application to the Evaluation of the Joint Torques

The inverse dynamic analysis is performed using the kinematic and dynamic data collected from the trial described. From this analysis, the net moments of force developed by the joint actuators at the joints of the biomechanical model are calculated. In particular, the net moments of force obtained for the joint actuators of the right ankle and knee joints are presented in Fig.16.18. The net moments-of-force, for the joints represented, are within the expected values reported in the literature for a normal cadence stride period [22]. The joint torques in all other anatomical joints is also obtained in the process, but not represented here for the sake of conciseness.

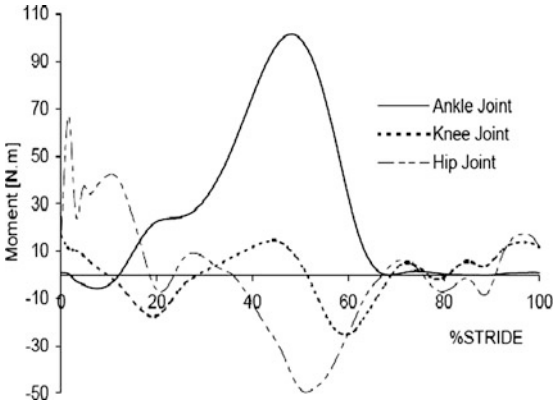



**Fig. 16.18** Net moment-of-force, i.e., torques, for the right ankle, knee and hip

Note that the motion generated by the distributed and hierarchical controller can be viewed as the motion acquired with the setup described. In this case, being the inverse dynamic analysis performed, the net moments-of-force obtained would be the joint torques developed by the low-level controllers. Therefore, in what follows, this equivalency is used to emphasize the relations between the muscle forces and the joint torques, but by using the gait analysis in the process.

## *16.6.2 The Redundant Problem: Muscle Forces*

The evaluation of the physiological merits of the solutions obtained requires that the biomechanical models used for the skeletal system are appropriate and do not eliminate the possibility of different muscle activation patterns. The muscle system modeled must also be complete enough to represent the actions required for the task and the muscle model supporting the analysis must be able to represent the muscle dynamics. The model used for the muscle activation dynamics and the strategies used to solve the muscle force sharing problem are described here.

### 16.6.2.1 Muscle Apparatus for the Locomotion System

The muscle apparatus implemented in the biomechanical model must be able to represent the human motion being studied. For the human locomotion a muscle apparatus, with 35 muscle actuators can be used to simulate the right lower extremity, as depicted by Fig. 16.19. The physiological information regarding the muscle definition is obtained from the literature [25–27] and compiled in a muscle database. This information consists in the maximum isometric force, resting length,

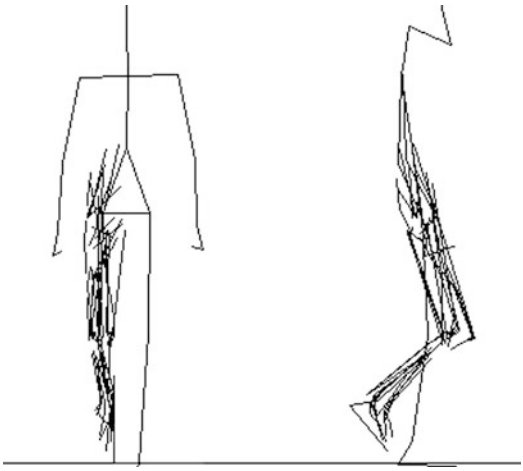


**Fig. 16.19** Muscles of the locomotion apparatus used in the biomechanical model

attachment points, wrap-around bodies and the local coordinates of the origin, insertion and via points.

The detailed muscle system model referred to and implemented in references by Silva [23] and by Yamaguchi [25] is not necessary for many application cases. In many study cases only part of the muscles referred are used in the supporting models, or different muscles are grouped together based on their functions.

#### 16.6.2.2 Hill-Type Muscle Model

The Hill muscle model, applied to the simulation of the muscle contraction dynamics, depicted in Fig. 16.20, is composed of a contractive element (CE) and a passive element (PE) contributing to the total muscle force $F^m(t)$. A series elastic element (SEE), associated with cross-bridge stiffness, is not included as it can be neglected in coordination studies not involving short-tendon actuators [28].

In the Hill muscle model, the contractile properties of the muscle tissue are controlled by its current length $l^m(t)$, rate of length change $\dot{l}^m(t)$ and activation $a^m(t)$. The force produced by the active Hill contractile element, for muscle $m$, is.

$$F^m_{CE}(a^m(t), l^m(t), \dot{l}^m(t)) = \frac{F^m_l(l^m(t))F^m_{\dot{l}}(\dot{l}^m(t))}{F^m_0} a^m(t) \tag{16.28}$$

where $F^m_0$ is the maximum isometric force and $F^m_l(l^m(t))$ and $F^m_{\dot{l}}(\dot{l}^m(t))$ represent the muscle force-length and force-velocity dependency [24, 28]

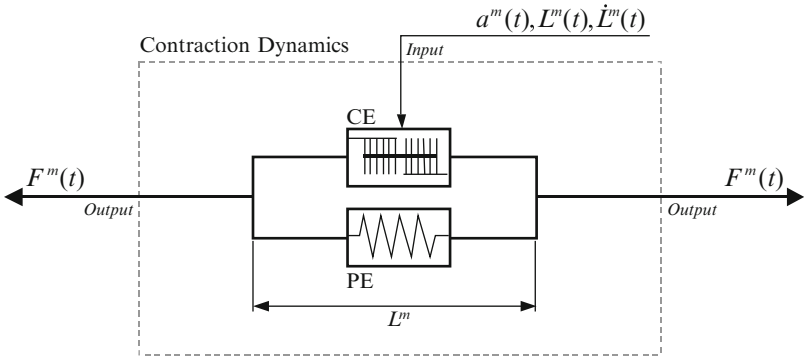



**Fig. 16.20** Contraction dynamics using a hill-type muscle model

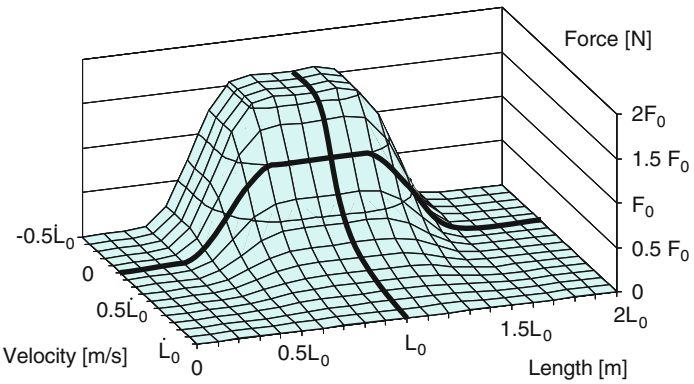



**Fig. 16.21** Carpet plots of the force-length and force-velocity relationships for the contractile element

$$F_l^m(l^m(t)) = F_0^m e^{-\left[\left[-\frac{9}{4}\left(\frac{l^m(t)}{l_0^m}-\frac{19}{20}\right)\right]^4-\frac{1}{4}\left[-\frac{9}{4}\left(\frac{l^m(t)}{l_0^m}-\frac{19}{20}\right)\right]^2\right]} \tag{16.29}$$

$$F_l^m(\dot{l}^m(t)) = \begin{cases} 0 & -\dot{l}_0^m > \dot{l}^m(t) \\ -\frac{F_0^m}{\arctan(5)}\arctan\left(-5\frac{\dot{l}^m(t)}{\dot{l}_0^m}\right)+F_0^m & 0.2\,\dot{l}_0^m \geq \dot{l}^m(t) \geq -\dot{l}_0^m \\ \frac{\pi F_0^m}{4\arctan(5)}+F_0^m & \dot{l}^m(t) \geq 0.2\,\dot{l}_0^m \end{cases} \tag{16.30}$$

where $l_0^m$ is the muscle resting length and $\dot{l}_0^m$ is the maximum contractile velocity, above which the muscle cannot produce force. A carpet plot of the force-length and force-velocity relations is presented in Fig. 16.21.

The passive element is independent of the activation and it only starts to produce force when stretched beyond its resting length $l_0^m$. The force produced by the passive element is approximated by [24]

$$F_{PE}^m(l^m(t)) = \begin{cases} 0 & l_0^m > l^m(t) \\ 8\frac{F_0^m}{l_0^{m^3}}\left(l^m - l_0^m\right)^3 & 1.63l_0^m \geq l^m(t) \geq l_0^m \\ 2F_0^m & l^m(t) \geq 1.63l_0^m \end{cases} \tag{16.31}$$

A muscle actuator equation associated to each contractile element is accomplished multiplying each actuator equation by a proper scalar factor. The Lagrange multiplier associated to such actuator is the muscle force or activation.

### 16.6.2.3 Redundant Muscle Force Solution by Optimization

The internal system of forces in the human body is highly redundant due to the large number of muscles that can team up in different ways to achieve the same motion. It can be argued that the muscle force sharing has the objective of fulfilling some criteria of optimality, referred here as $\mathcal{F}_0(\mathbf{a})$. Being that the case, the corresponding optimization problem is formulated mathematically as

$$\begin{aligned} &\text{minimize } \mathcal{F}_0(\mathbf{a}) \\ &\text{subject to } \left\{ \left[\mathbf{\Phi}_{\mathbf{q}}^T - \mathbf{C}^T\right] \begin{Bmatrix} \boldsymbol{\lambda} \\ \mathbf{f}_{unknown} \end{Bmatrix} - \{-\mathbf{M}\ddot{\mathbf{q}} + \mathbf{g}_{known}\} = \mathbf{0} \right. \\ &\qquad\qquad \mathbf{0} \leq \mathbf{a} \leq 1 \end{aligned} \tag{16.32}$$

where $\mathbf{f}_{unknown} = \mathbf{f}(\mathbf{a})$ vector represents the history of muscle forces for all muscles and for all time frames, described by Hill model [29].

In the optimization problem the muscle forces are often used as design variables [10, 14]. Due to the relatively large range of variation of the muscle forces these variables often lead to problems for the optimizer algorithms to converge, leading to erroneous results. Taking into account the muscle contraction dynamics, described by the Hill muscle model, all quantities involved depend solely on the muscle physiology, on the kinematics of the muscle and on its activation, which is within the range [0,1]. Because the muscle physiological parameters are known beforehand and the kinematic quantities, length and velocity, are directly associated to the state variables of the biomechanical model the only unknowns are the muscle activations.

A procedure to discretize activation profiles is to sample the muscle activation functions at given points, the time frames of the inverse dynamic analysis, and interpolating the values between sampling points by piecewise linear functions, leading to

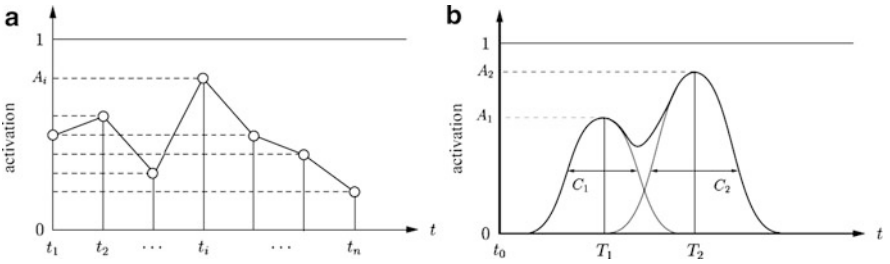



**Fig. 16.22** Functions representing muscle activation profiles: (**a**) piecewise linear function; (**b**) a two-bump activation function using the $C^\infty$ approach

$$a(t) = A_i + (t - t_i)\frac{A_{i+1} - A_i}{t_{i+1} - t_i}, \quad i = 0, \ldots n; \quad t \in [t_i, t_{i+1}[ \tag{16.33}$$

where $t_i$ are the time frames, $A_i$ are the sampling values and $a(t)$ is the resulting activation function. By using the sampled values $A_i$ as design parameters, the optimization routine can find the best fit. This approach, shown in Fig. 16.22a, is called input sampling and leads to a large number design parameters (number of muscles $\times$ number of samples) can pose problems to the convergence of the optimizer [30].

Another approach, proposed by Strobach et al. [30], employs a limited number of exponential functions that render smooth bump behavior. The goal of the optimization is to determine location, amplitude and width of these exponential bumps, and therefore using them as design parameters. For a $n$-bump activation profile the function is

$$a(t) = A_1\, e^{-C_1(t-T_1)^2} + A_2\, e^{-C_2(t-T_1)^2} + \ldots + A_n\, e^{-C_n(t-T_1)^2} \tag{16.34}$$

This approach is denoted as $C^\infty$ and its principle is depicted in Fig. 16.22b.

Notice that experimental observations of muscle activation patterns, obtained by EMG, show that for a given task only a limited number of bumps exist, generally one or two. Assuming that each muscle has two bumps for the activation function, the number of design variables for the $C^\infty$ approach is $2 \times 3$ bump properties $\times$ number of muscles, which is a much smaller number of design variables than in the discrete activation profile approach.

Using a piecewise linear function, the design variables used in this work to represent the muscle activation of the $m$ muscles in the $n$ time steps are arranged in a $mn \times 1$ vector, called vector of global muscle activation, arranged as

$$\mathbf{a} = \left[a_1^1, \ldots, a_j^1, \ldots, a_n^1, \ldots, a_1^i, \ldots, a_j^i, \ldots, a_n^i, \ldots, a_1^m, \ldots, a_j^m, \ldots, a_1^m\right]^T \tag{16.35}$$

being $a_j^i$ the activation of the $i^{th}$ muscle for time $t_j$.

#### 16.6.2.4 Objective Functions for Biomechanics

The most common objective functions used in biomechanics are those associated to physiological criteria. The idea is that a cost function must reflect the inherent physical activity or pathology and to include relevant physiological characteristics and functional properties, such as the maximum isometric force or the electromyographic activity [31]. The sum of the square of the individual muscle forces fulfills the objective of energy minimization. This cost-function, which does not include physiological or functional capabilities, is defined as [31]

$$\mathcal{F}_0(\mathbf{a}) = \sum_{m=1}^{n_{ma}} \left(F_{CE}^m\right)^2 \tag{16.36}$$

Crowninshield and Brand [13] introduced a function based on a quantitative force-endurance relationship and on experimental results, which includes physiological information, namely the value of the physiological cross sectional area of each muscle. This function, reported to predict co-activation of muscle groups in a physiologically realistic manner, is given by

$$\mathcal{F}_0(\mathbf{a}) = \sum_{m=1}^{n_{ma}} \left(\sigma_{CE}^m\right)^3 \tag{16.37}$$

Developments in the study of human gait suggest that metabolic cost models lead to better predictions of human gait patterns [32]. Based on the work by Umberger et al. [32] Schiehlen and Ackerman [33] propose an objective function that describes the metabolic costs of transport for all $n$ muscles involved in a given gait cycle as

$$\mathcal{F}_0(\mathbf{a}) = \frac{1}{\Delta S} \sum_{i=1}^{n} \int_{t_0}^{t_e} \dot{E}\left(u_i(t), a_i(t), v_i^{CE}(t), l_i^{CE}(t), f_i^{CE}(t), p_i\right) dt \tag{16.38}$$

where $\Delta S$ is the distance traveled in the gait cycle, $[t_0, t_e]$ the period of time for the gait cycle, sampled into $m$ instants, $\dot{E}$ the total metabolic energy, $u_i(t)$ represents the muscle neural excitation, $a_i(t)$ is the muscle activation function, $v_i^{CE}(t)$ the muscle shortening velocity, $l_i^{CE}(t)$ the muscle length, $f_i^{CE}(t)$ the muscle force and $p_i$ represents all parameters required for muscle $i$ in the Hill model.

#### 16.6.2.5 Numerical Solution of the Optimization Problem

In contrast with the traditional time-independent static optimization methods that do not take into account the entire motion, i.e., each time discretization is independently solved [34, 35], in case of using the metabolic cost objective function

depicted by (16.38) the evaluation of the muscle forces is done for the complete motion data at the same time. This implies that the optimization problem, initially depicted by (16.32), is now re-written as

$$\begin{aligned} \min \mathcal{F}_0(\mathbf{a}) &= \frac{1}{\Delta S}\sum_{i=1}^{n}\int_{t_0}^{t_e}\dot{E}\left(u_i(t),a_i(t),v_i^{CE}(t),l_i^{CE}(t),f_i^{CE}(t),p_i\right)dt \\ \text{sub. to} &\begin{cases} \left(\left[\boldsymbol{\Phi}_{\mathbf{q}}^T-\mathbf{C}^T\right]\begin{Bmatrix}\boldsymbol{\lambda}\\ \mathbf{f}_{unknown}\end{Bmatrix}-\{-\mathbf{M}\ddot{\mathbf{q}}+\mathbf{g}_{known}\}\right)_t=0,\ t=1,...,n \\ 0\le\mathbf{a}_t\le 1,\ t=1,...,n \end{cases} \end{aligned} \tag{16.39}$$

The optimization method selected to solve the muscle force sharing problem is based on the use of a sequential quadratic programming method (SQP), such as the one implemented in the DOT library [36]. In the computational implementation of the optimization procedure it is recognized that the equality constraints equations represent a set of linear equations on the control variables, represented by the vector of Lagrange multipliers. Hence, the gradients of these equations can be obtained analytically as:

$$\nabla_{\lambda}\mathbf{f}_n^m=\frac{\partial\mathbf{f}}{\partial\boldsymbol{\lambda}}=\boldsymbol{\Phi}_{\mathbf{q}}^T \tag{16.40}$$

This result reveals that there is no need to calculate the sensitivities by finite differences or any other numerical method, because these quantities are already known, in analytical form, needing only to be introduced in the optimization process, without any further calculation.

The dimension of the optimal problem includes $m\times n$ variables, $n_{dof}\times n$ equality constraints and $m\times n$ inequality constraints. Clearly, the size of the problem is proportional to the number of time frames used in the analysis, which can put serious challenges to the optimization methods that can be selected to solve the problem.

#### 16.6.2.6 Results of the Application to the Evaluation of the Muscle Forces

The data collected for the gait case depicted in Fig. 16.17 is used to evaluate the muscle forces and the anatomical joint reaction, using the optimization problem described by (16.39). The results are fully described in the work by Rodrigo et al. [15] being only a selected number of muscles shown in Figs. 16.23, 16.24, and 16.25. By observing the muscle forces obtained it is noticed that the muscular control of the hip joint, for which the net-moments of force are shown in Fig.16.18, during stance period is fundamentally controlled by the extensor (*Gluteus Maximus, Semimembranosus, Semitendinosus* and *Aductor Magnus*) and abductor muscles

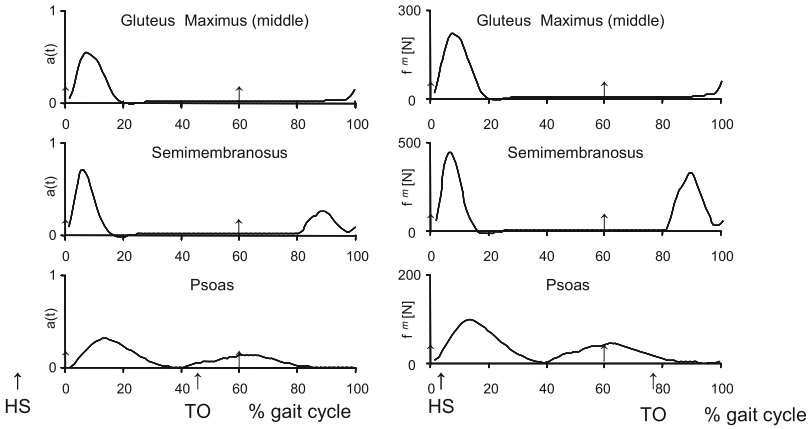



**Fig. 16.23** Activation (*left*) and force (*right*) patterns for selected muscles of the right leg associated to the hip joint (HS – Heel Strike, TO – Toe-Off)

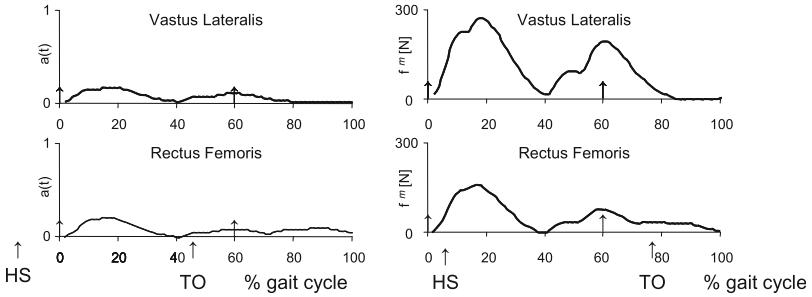



**Fig. 16.24** Activation (*left*) and force (*right*) patterns for selected muscles of the right leg associated to the knee joint (HS – Heel Strike, TO – Toe-Off)

(*Gluteus Maximus, Medius* and *Minimus* and *Tensor Fasciae Latae*),while during oscillation, flexor muscles do the job (*Aductor Longus, Brevis and Magnus, Psoas, Ilíaco, Sartorius* and *Rectus Femoris*). Furthermore, the participation of adductor muscles is limited to the transition period between stance and oscillation (*Aductor Longus, Brevis* and *Magnus* and *Quadratus Femoris*).

Diverse muscles selectively supervise the muscular knee joint control throughout gait cycle, rendering the stability and mobility necessary to walk while remaining inactive as long as possible, to follow an energy conservation strategy. Among all these muscles, some of them get involved exclusively on knee joint, as is the case of the *Vastus Intermedius Lateralis* and *Medialis* and *Rectus Femoris,* as extensor muscles. The remaining muscles flex the knee while supervising the hip joint control, i.e., *Bíceps Femoris, Gracilis, Sartorius, Semimembranosus, Semitendinosus, Tensor Fasciae Latae*. Also linked to the knee, though mainly acting on ankle joint control, the *Gemelli* muscles help to flex the knee joint [23, 33].

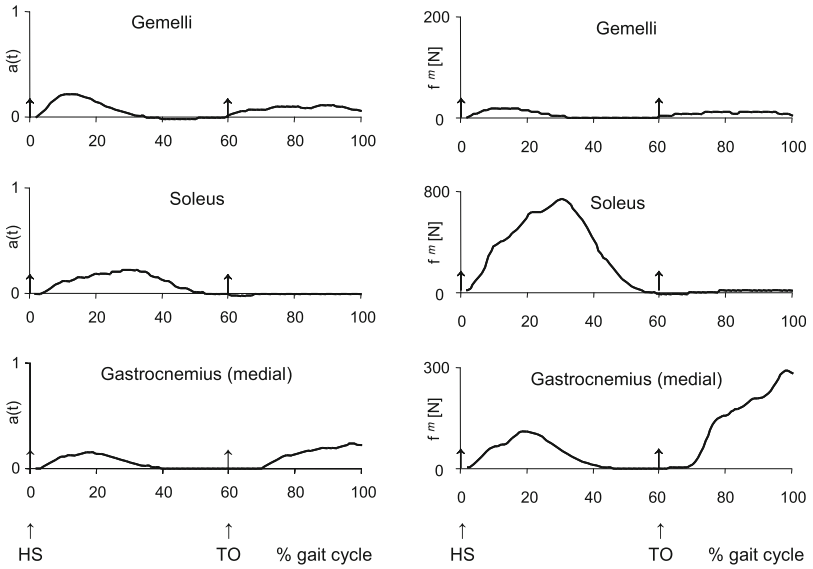



**Fig. 16.25** Activation (*left*) and force (*right*) patterns for selected muscles of the right leg associated to the ankle joint (HS – Heel Strike, TO – Toe-Off)

The muscle activity pattern of ankle joint is also discussed in view of the sequence of phases of the gait cycle. While the plantar flexor muscles mainly act during the stance phase through the *Flexor Digitorum Longus, Flexor Hallucis Longus, Gastrocnemius medial and lateral heads*, *Peroneus Brevis, Peroneus Longus, Soleus, Gemelli* and *Tibialis Posterior*, the dorsiflexor muscles operate at oscillation phase using the *Extensor Hallucis Longus, Extensor Digitorum Longus, Peroneus Tertius* and *Tibialis Anterior*, with exception of eccentric activity of dorsiflexor muscles, posterior to foot impact during the stance period of gait cycle. Among all plantar flexor muscles of ankle joint, the *Soleus* is the main flexor, accounting for about 93% of total capacity, in agreement with its function of shock absorptor due to the foot initial contact. The second most important muscles in flexing are the *Gemelli* muscle. Concerning the dorsiflexor muscles, the force intensity of *Tibialis Anterior* is larger than that of the other muscles.

The analysis of the muscle functions for each anatomical joint show a relation between the muscle forces and the joint kinematics. This type of analysis can be made for any kind of motion, provided that the proper musculoskeletal representation is implemented in the biomechanical model. Examples of the application of this type of approach to other parts of the human body are available in Refs. [26, 27].

## 16.7 Conclusions

The application of control strategies to human motion coordination motion planning requires not only that suitable biomechanical models are available but also that the internal forces developed by the musculoskeletal system are properly described. When the joint torques are used as representatives of the muscle forces the control of the human body requires that a controller is associated to each one of the biomechanical system degrees of freedom making the motion planning a difficult task to control. When the individual muscles of the human body have to be controlled independently, if physiologically reasonable, the control task would be virtually impossible. This work shows that by decentralizing and implementing a hierarchical structure in the control it is possible to devise a strategy to design human motion. The feasibility of the joint torques developed by the controllers can only be evaluated by estimating the muscle forces necessary to obtain the joint torques. Here, a methodology based on the use of a detailed musculoskeletal model, a representation of the muscle activation dynamics by the Hill muscle model and an optimization procedure is proposed to obtain the individual muscle forces and the corresponding joint torques. By uniquely associating the muscle forces and the joint torques to a particular motion, obtained by the control strategy outlined, it is possible to evaluate the motion feasibility. The procedures outlined can be further developed by using the muscle force sharing evaluation integrated in the control system associated to the low-level PD controller gains. Such strategy is the object of further developments of the work presented here.

## References

1. Ambrósio J, Lopes G, Silva M (1999) Reconstruction of the spatial motion of biomechanical systems by means of computer vision and multibody dynamics. In: Kecskemethy A, Schneider M, Woernle C (eds) Advances in multibody systems and mechatronics, Technische Univ. Graz, Institute fur Mechanik und Getriebelehere, Austria, pp 407–426
2. Zielinska T (2004) Biological aspects of locomotion. In: Pfeiffer F, Zielinska T (eds) Walking: biological and technological aspects. Springer, Wien, pp 1–29
3. Silva M, Ambrósio J, Pereira M (1997) Biomechanical model with joint resistance for impact simulation. Multibody Syst Dyn 1:65–84
4. Sandell N, Varaiya P, Athans M, Safonov M (1978) Survey of decentralized control methods for large scale systems. IEEE Trans Automat Contr 23(2):108–128
5. Tsai D, Arabyan A (1991) Decentralized and hierarchical control of articulated multibody systems. Technical Report CAEL-91-2, Computer-Aided Engineering Laboratory, Department of Aerospace and Mechanical Engineering, The University of Arizona, Tucson, Arizona
6. Russel S, Norvig P (1995) Artificial intelligence, a modern approach. Prentice Hall, Englewood-Cliffs
7. Garriott OK (2011) Conservation Laws in Zero G, ST0046; Opportunities in zero gravity, ST0046, Textbook Tapes, Inc. http://www.textbooktapes.com
8. Kane TR, Headrick MR, Yatteau JD (1972) Experimental investigation of an astronaut maneuvering scheme. J Biomech 5:313–320

9. Kane TR, Scher MP (1970) Human self-rotating by means of limb maneuvers. J Biomech 3:39–49
10. Seireg A, Arvikar R (1989) Biomechanical analysis of the musculoskeletal structure for medicine and sports. Hemisphere, New York
11. Buchanan TS, Shreeve DA (1996) An evaluation of optimization techniques for the prediction of muscle activation patterns during isometric tasks. J Biomech Eng 118:565–574
12. Tsirakos D, Baltzopoulos V, Bartlett R (1997) Inverse optimization: functional and physiological considerations related to the force-sharing problem. Crit Rev Biomed Eng 25(4–5):371–407
13. Crowninshield R, Brand R (1981) Physiologically based criterion of muscle force prediction in locomotion. J Biomech 14(11):793–801
14. Ambrósio J, Kecskeméthy A (2007) Multibody dynamics of biomechanical models for human motion via optimization. In: Orden JC et al (eds) Computational multibody dynamics. Springer, Dordrecht, pp 245–272
15. Rodrigo S, Ambrósio J, Silva M, Penisi O (2008) Analysis of human gait based on multibody formulations and optimization tools. Mech Based Des Struct Mach 36(4):446–477
16. Nikravesh P (1988) Computer-aided analysis of mechanical systems. Prentice Hall, Englewood-Cliffs
17. Silva M, Ambrósio J (2002) Kinematic data consistency in the inverse dynamic analysis of biomechanical systems. Multibody Syst Dyn 8:219–239
18. Ambrósio J, Silva M (2007) Methodologies for forward and inverse dynamic analysis of the biomechanics of human motion. In: Gonzalez Y, Cerrolaza M (eds) Bioengineering modeling and computer simulation. CIMNE, Barcelona, pp 13–33
19. Ambrósio J (2009) Human spatial attitude control by means of zero momentum turns. Int J Comput Vis Biomech 2(2):171–176
20. Kuo B (1995) Automatic control systems. Prentice Hall, Englewood-Cliffs
21. Silva M, Gonçalves J (1995) APOLLO, 3D dynamic analysis of mechanisms. IDMEC, Instituto Superior Técnico, Portugal
22. Winter DA (1990) Biomechanics and motor control of human movement. John Wiley & Sons, New York
23. Silva M, Ambrósio J (2003) Solution of the redundant muscle forces in human locomotion with multibody dynamics and optimization tools. Mech Based Des Struct Mech 31(3):381–411
24. Ambrósio J, Silva M (2005) A biomechanical multibody model with a detailed locomotion muscle apparatus. In: Ambrósio J (ed) Advances in computational multibody systems. Springer, Dordrecht, pp 155–184
25. Yamaguchi G (2001) Dynamic modeling of musculoskeletal motion. Kluwer Academic, Boston
26. Garner BA, Pandy MG (2001) Musculoskeletal model of the upper limb based on the visible human male dataset. Comput Methods Biomech Biomed Eng 4(2):93–126
27. Quental C, Folgado J, Ambrósio J, Monteiro J (2012) Dynamic analysis of a multibody system of the upper limb. Multibody Syst Dyn, 28(1-2):83–108. DOI 10.1007/s11044-011-9297-0
28. Zajac F (1989) Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Crit Rev Biomed Eng 17(4):359–411
29. Winters J (1995) Concepts in neuromuscular modelling. In: Allard P, Stokes I, Blanchi JP (eds) Three-dimensional analysis of human movement. Human Kinetics, Champaign
30. Strobach D, Kecskeméthy A, Steinwender G, Zwick B (2005) A simplified approach for rough identification of muscle activation profiles via optimization and smooth profile patches. In: Goicolea J et al (eds) Proceedings of multibody dynamics 2005, ECCOMAS thematic conference, Madrid, Spain, 21–24 June 2005, pp 1–17
31. Tsirakos D, Baltzopoulos V, Bartlett R (1997) Inverse optimization: functional and physiological considerations related to the force-sharing problem. Crit Rev Biomed Eng 25 (4–5):371–407
32. Umberger BR, Gerritsen KGM, Martin PE (2003) A model of human muscle energy expenditure. Comput Methods Biomech Biomech Eng 6(2):99–111

33. Schiehlen W, Ackerman M (2005) Estimation of metabolical costs for human locomotion. In: Proceedings of ASME international design engineering technical conferences and computers and information in engineering conference 2005, Technical Paper DETC2005-842229, Long Beach, CA, 24–28 Sep 2005
34. Czaplicki A, Silva M, Ambrósio J, Jesus O, Abrantes J (2006) Estimation of the muscle force distribution in ballistic motion based on multibody methodology. Comput Methods Biomech Biomech Eng 9(1):45–54
35. Czaplicki A, Silva M, Ambrósio J (2004) Biomechanical modelling for whole body motion using natural coordinates. J Theor Appl Mech 42(4):927–944
36. Vanderplaats G (1999) DOT – Design Optimization Tools – Users Manual – Version 5.0

# Chapter 17
# Application Examples of Wire Robots

**Tobias Bruckmann, Wildan Lalo, and Christian Sturm**

**Abstract** Wire Robots have become both a wide research field as well as a promising subject to application projects. The Chair of Mechatronics at the University Duisburg-Essen has been successful in the setup of prototypes in several application fields. Within this paper, two projects taking advantage from the special properties of wire robots are presented: The first project aims at the development of a wind tunnel suspension system. The second project focuses on the realization of a revolutionary storage and retrieval machine for high racks. Using the mechatronic approach of simulation-based development, major aspects of modeling, simulation, design, trajectory planning and practical realization are discussed.

## 17.1 Introduction

Parallel kinematic machines have major advantages compared to serial manipulators in terms of precision, load distribution and stiffness. On the other hand, classical parallel kinematics (e.g. the Stewart-Gough platform) have a relatively small workspace compared to serial systems of the same overall construction volume. In 1985, Landsberger and Sheridan [14] presented the concept of a parallel tendon-based manipulator where the conventional linear actuators (e.g. hydraulic cylinders) are replaced by wires and motor-driven winches. This leads to a number of interesting properties, avoiding the drawbacks of the classical parallel kinematics:

T. Bruckmann (✉) • W. Lalo • C. Sturm
Chair of Mechatronics, University Duisburg-Essen, Duisburg and Essen,
North Rhine-Westphalia, Germany
e-mail: tobias.bruckmann@uni-due.de; wildan.lalo@uni-due.de; christian.sturm@uni-due.de

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_17, 

- The flexibility of wires allows very large changes in the length of the kinematic chain, for example by coiling the wire on a winch drum. Therefore, the workspace of a wire robot may become extremely large. As an example, wire-suspended cameras [2] may span a complete sports stadium.
- The coiling can be done by extremely fast motor-driven winches. Usually, the loads are comparably lightweight. Since the moving mass of the wire robot is nearly negligible, very high accelerations and velocities can be realized.
- The mechanical components needed are simple and of industrial grade and maturity. No complex joints are used.
- A wire is a unilateral constraint. Therefore, wires can transmit only tension forces and thus at least $m = n + 1$ wires are needed to tense a system having $n$ degrees-of-freedom [16].

Wire-driven parallel robots (or simply wire robots) have made a long way from pure fundamental research—mainly done by the very active community of researchers working on kinematic analysis and design—and are now subject to numerous application projects (e.g. EU-project CableBOT: Parallel Cable Robotics for Improving Maintenance and Logistics of Large-Scale Products).

At the Chair of Mechatronics, already two application projects based on wire robot technology could be realized: The first project was done in close cooperation with experts on wind tunnel experiments. Here, the development of the design and the control concept are described. The second project is a large public-funded project where a revolutionary storage and retrieval machine for high racks is developed. This project aims at the realization of a demonstrator within a fully working intralogistics environment and therefore, components of industrial grade are used. Within this project, the geometrical design, the design engineering, the control concept and the virtual prototyping have been subject to intensive work.

## 17.2 Dynamics Modeling of Wire Robots

Taken from [6], the platform pose $\boldsymbol{x} = \left[ {}^{B}\boldsymbol{r}^{T}\ \varphi\ \vartheta\ \psi \right]$ and the base points ${}^{B}\boldsymbol{b}_i$, $i = 1 \leq i \leq m$ are referenced in the inertial frame $\mathcal{B}$. Contrarily, the platform connection points $\boldsymbol{p}_i$ are referenced in the platform-fixed coordinate frame $\mathcal{P}$. The orientation of the platform referenced in the base frame is defined by the rotation matrix ${}^{B}\boldsymbol{R}_{P}$. For simplicity, Roll-pitch-yaw angles are used.

Assuming the wires are led by point-shaped guidances (e.g. small ceramic eyes) from the winches to the platform, the base vectors ${}^{B}\boldsymbol{b}_i$ are constant. Now the vector chain pictured in Fig. 17.1a delivers

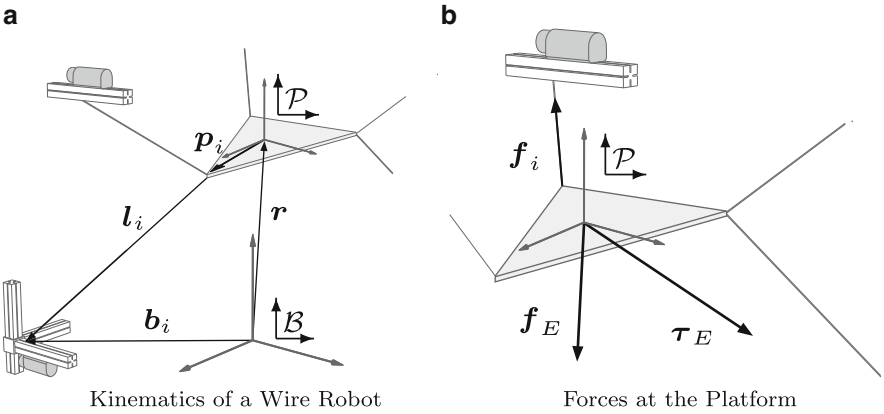



**Fig. 17.1** Simple modeling of a wire robot

$$ {}^B\boldsymbol{l}_i = {}^B\boldsymbol{b}_i - \underbrace{{}^B\boldsymbol{r} - {}^B\boldsymbol{R}_P\,{}^P\boldsymbol{p}_i}_{{}^B\boldsymbol{p}_i}, \quad 1 \le i \le m. \tag{17.1} $$

immediately. Hence, the length of the $i$th wire can be calculated by

$$ l_i = \left\| {}^B\boldsymbol{b}_i - {}^B\boldsymbol{p}_i \right\|_2, \quad 1 \le i \le m. \tag{17.2} $$

Assuming a base frame $\mathcal{B}$ and a platform-fixed frame $\mathcal{P}$, the accumulated wrench $\boldsymbol{w}_{wire}$ of the wires forces $\boldsymbol{f}$ acting onto the platform can be written as

$$ \boldsymbol{w}_{wire} = [\boldsymbol{f}_{wire}\, \boldsymbol{\tau}_{wire}]^T = \left[ \sum_{i=1}^{m} \boldsymbol{f}_i \sum_{i=1}^{m} \boldsymbol{p}_i \times \boldsymbol{f}_i \right]^T. \tag{17.3} $$

Since the forces act along the wires

$$ \boldsymbol{f}_i = f_i \cdot \frac{\boldsymbol{l}_i}{\|\boldsymbol{l}_i\|_2} = f_i \cdot \boldsymbol{\nu}_i, \quad (1 \le i \le m) \tag{17.4} $$

holds. It follows

$$ \boldsymbol{w}_{wire} = \begin{bmatrix} \boldsymbol{\nu}_1 & \cdots & \boldsymbol{\nu}_m \\ \boldsymbol{p}_1 \times \boldsymbol{\nu}_1 & \cdots & \boldsymbol{p}_m \times \boldsymbol{\nu}_m \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \boldsymbol{f} = \boldsymbol{A}^T \boldsymbol{f}. \tag{17.5} $$

The Newton-Euler equations lead to

$$ m_p \ddot{\boldsymbol{r}} = \boldsymbol{f}_E + \boldsymbol{f}_{wire} \tag{17.6} $$

$$\boldsymbol{I}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times (\boldsymbol{I}\boldsymbol{\Omega}) = \boldsymbol{\tau}_{wire} + \boldsymbol{\tau}_E, \tag{17.7}$$

with

$m_p$: the mass of platform,
$\boldsymbol{I} \in \mathbb{R}^{3\times3}$: inertia tensor defined with respect to the inertial system $\mathcal{B}$ which is an expression of rotation angles,
$\boldsymbol{\Omega}$: angular velocity of the platform in $\mathcal{B}$,
$\boldsymbol{f}_E$: vector of external forces,
$\boldsymbol{\tau}_E$: vector of external torques.

Equation 17.6 can be rewritten by

$$\underbrace{\underbrace{\begin{bmatrix} m_p\boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{M}_p} \underbrace{\begin{bmatrix} \ddot{\boldsymbol{r}} \\ \dot{\boldsymbol{\Omega}} \end{bmatrix}}_{\ddot{\boldsymbol{x}}} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Omega} \times (\boldsymbol{I}\boldsymbol{\Omega}) \end{bmatrix}}_{\boldsymbol{g}_C} - \underbrace{\begin{bmatrix} \boldsymbol{f}_E \\ \boldsymbol{\tau}_E \end{bmatrix}}_{\boldsymbol{g}_E}}_{-\boldsymbol{w}} = \boldsymbol{A}^T\boldsymbol{f}. \tag{17.8}$$

with

$\boldsymbol{M}_p$: mass matrix of platform,
$\boldsymbol{E}$: indentity matrix,
$\boldsymbol{g}_C \in \mathbb{R}^{n\times1}$: Cartesian space vector of Coriolis and centrifugal forces and torques,
$\boldsymbol{g}_E \in \mathbb{R}^{n\times1}$: vector of the generalized applied forces and torques, not including the resultants of wire tensions.

Taking wire minimum and maximum force limits $\boldsymbol{f}_{min}$ and $\boldsymbol{f}_{max}$, respectively, into account [5] it follows

$$\boldsymbol{A}^T\boldsymbol{f} + \boldsymbol{w} = 0 \text{ with} \tag{17.9}$$

$$\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \tag{17.10}$$

Obviously (17.9) represents an underdetermined system of linear equations. Its solution space is $r = m - n$-dimensional. Hence isolating the force distribution $\boldsymbol{f}$ leads to

$$\boldsymbol{f} = -\boldsymbol{A}^{+\mathrm{T}}\boldsymbol{w} + \boldsymbol{H}\,\boldsymbol{\lambda}, \tag{17.11}$$

where $\boldsymbol{A}^{+\mathrm{T}}$ denotes the Moore-Penrose Pseudo-Inverse of $\boldsymbol{A}^T$. In other words, the task of finding a feasible wire force distribution is equivalent to the task of finding $\boldsymbol{\lambda} \in \mathbb{R}^r$ such that $\boldsymbol{f} > f_{min}$, with $f_{min} > 0$ holds. Note that $\boldsymbol{H}$ is the nullspace or kernel of $\boldsymbol{A}^{\mathrm{T}}$ defined as

$$\boldsymbol{H} := [\boldsymbol{h}_1 \dots \boldsymbol{h}_r], \tag{17.12}$$

where

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{h}_i = \boldsymbol{0}, \quad 1 \leq i \leq r. \tag{17.13}$$

## 17.3 Realization of a Wind Tunnel Suspension System

The concept of using a wire robot suspension system in wind tunnels was proposed by Lafourcade et al. [13] and Yaqing and Xiongwei [24]. In their applications, the very small thickness of the wires turned out to be extremely advantageous since the air flow in the windtunnel is not disturbed. This is important since the suspension system should not influence the results of the aerodynamical experiments performed. Noteworthy, in their applications the mass of the moved airplane models was comparably low.

In the project presented here, the original parallel wire robot concept is modified: Instead of winches and wires, linear actuators and rail systems are used while tendons of fixed length connect the skids on the rails with the end effector. This leads to a kinematical structure comparable to a conventional parallel manipulator with additional redundant legs and results in a kinematical redundancy.

The project is done in close cooperation with the Technical University Hamburg-Harburg, where ship models must be moved on defined trajectories within a wind tunnel. The experiments require the motion of heavyweight payloads up to 100 kg with a frequency of up to 0.5 Hz for the translational degrees-of-freedom and up to 2.5 Hz for the rotational degrees-of-freedom. This demands for a high drive power which is additionally raised by the need for internal tension. Therefore, an energy-saving concept had to be found.

Additionally, a wide range of rotational motions should be possible to realize arbitrary maneuvers. This leads to the problem of finding an adequate geometry design. Due to architectural limitations in the wind tunnel, the geometry of the supporting frame is fixed.

In [20] two different design concepts were investigated. The first one uses a rail-based system with eight wires of constant length. The configuration of this mechanism is shown in Fig. 17.2. Each skid is connected to the model by a wire of constant length. Although two skids share one rail, every skid is separately operated by a SEW EURODRIVE DC motor via a belt drive. Noteworthy, the use of linear drives for wire robots was introduced by Merlet [15] who proposed that idea both for very high loads and extremely high accelerations and velocities using pulleys. The second concept is based on the common and well known motor winch approach as it is used e.g. at the SEGESTA-prototype of the University Duisburg-Essen in Duisburg, Germany [10], or the IPANEMA-prototype of the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany [17].

A comparison of the characteristics of both systems has shown, that the winch based system provides a larger translational workspace, while the rail-based system

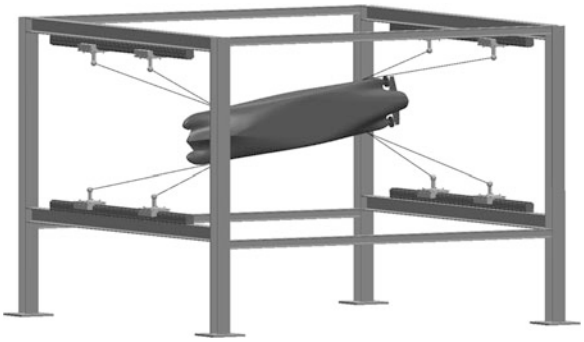


**Fig. 17.2** Principle of application

allows for the use of motors with a considerably lower power consumption. For the application, the rotational movements of the model are of larger interest. Therefore, although having a smaller workspace, the rail-based still affords a wide range of ship maneuvers. Due to the lower power consumption a decision for the rail-based system was made. Additionally, due to its intrinsic modularity, the suspension system installed can be easily adapted to the specific needs of each experiment.

### *17.3.1 Modeling of the Rail-based System*

According to [20] the kinematics of the introduced system is given by

$$q_i = -(\boldsymbol{r}_{S_i} - \boldsymbol{r}_{C_i})\boldsymbol{n}_{Ri} + (-1)^i \sqrt{((\boldsymbol{r}_{S_i} - \boldsymbol{r}_{C_i})\boldsymbol{n}_{Ri})^2 - (\boldsymbol{r}_{S_i} - \boldsymbol{r}_{C_i})^2 + l_i^2} \qquad (17.14)$$

with

$q_i$: position of the skid i,
$l_i$: length of wire i,
$\boldsymbol{n}_{Ri}$: unit vector in direction of the rail i,
$\boldsymbol{r}_{Ci}$: vector to *i*th wire attachment point on the ship model
$\boldsymbol{r}_{Si}$: vector to a reference point on rail i,

The dynamical equations of motion of the end effector have been introduced in Sect. 17.2. Due to the fact that the presented robot is twofold redundant, (17.8) represents an under-determined system of linear equations. Therefore the calculation of the wire force distribution is not straightforward. Fast motions of the end effector are desired to be generated. According to that fact a minimum positive solution of the wire force distribution is desired in order to reduce the motor power consumption and the applied load on the mechanical components. Additionally, the unilateral properties of the wires have to be taken into account: On the one hand, wires have a limited breaking load, on the other hand, the wires need a defined minimum tension

to avoid slackness. The force distribution can be formulated as a constrained quadratic optimization problem with

$$minimize \sum_{i=1}^{m} f_i^2$$

$$s.t. \quad \boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \quad \wedge \quad \boldsymbol{A}^T\boldsymbol{f} + \boldsymbol{w} = \boldsymbol{0}. \tag{17.15}$$

For the application, a real-time capable active set method has been used to solve the quadratic optimization problem.

Each wire is driven by a combination of a skid and a DC motor. For the dynamical equations of the skid subsystem

$$\boldsymbol{M}_s\ddot{\boldsymbol{q}} + \boldsymbol{D}_s\dot{\boldsymbol{q}} + \boldsymbol{T}\boldsymbol{f} = \boldsymbol{f}_s \tag{17.16}$$

with

$\boldsymbol{M}_s$: inertia matrix of the skids,
$\boldsymbol{D}_s$: diagonal matrix of coulomb friction between skid and rail,
$\boldsymbol{q}$: vector of the skid positions,
$\boldsymbol{T}$: diagonal matrix that projects the wire force components into the direction of the skid movement,
$\boldsymbol{f}_s$: skid driving force vector

holds. Motor and skid are connected by a gear belt. The elasticity of the wires are considered as a massless spring-damper system. The dynamical equations of the DC motors can be described by

$$\boldsymbol{M}_m\ddot{\boldsymbol{\theta}} + \boldsymbol{D}_m\dot{\boldsymbol{\theta}} + \eta\boldsymbol{f}_s = \boldsymbol{u} \tag{17.17}$$

with

$\boldsymbol{M}_m$: inertia matrix of the drive units including crown gear and motor,
$\boldsymbol{D}_m$: diagonal matrix of coulomb friction at the crown gear bearing,
$\eta$: radius of the crown gear,
$\boldsymbol{\theta}$: vector of motor shaft angles,
$\boldsymbol{u}$: electromechanical driving torque vector.

By use of the Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}} \tag{17.18}$$

of the manipulator, with the dynamics of the whole system can be derived in the form

$$\boldsymbol{M}_{eq}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{N}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = -\boldsymbol{A}^T \boldsymbol{T}^{-1} \boldsymbol{\tau} = \boldsymbol{F}_v \tag{17.19}$$

with

$\boldsymbol{M}_{eq}$: inertia matrix of the whole system,
$\boldsymbol{N}$: vector containing all velocity dependent terms and external forces,
$\boldsymbol{F}_v$: virtual force wrench to drive the system due to a given trajectory.

### 17.3.2 Control

The most common control approach for wire based parallel kinematic system allows for the use of a PD joint control of the individual actuator while having rather small end effector loads. Within the introduced application ship models with a weight up to 100kg are going to be actuated. Due to the comparably high payload, this approach is not sufficient. By only controlling the actuator joints, $q_i$ in this case, end effector and wire forces are not part of the controller loop. Experiments with the prototype SEGESTA have shown, that this approach still provides good results with respect to end effector pose error, but leads to unrequested, since uncontrolled, high wire forces. The intended approach has first been introduced by [23]. Another control loop, that uses a model of the inverse dynamics of the whole system, is superposed to the common joint control law. With

$$\boldsymbol{F}_v = \boldsymbol{M}_{eq}(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{N}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{17.20}$$

a model of the system dynamics is used to linearize around the current working point. Substituting (17.19) with (17.20) leads to

$$\ddot{\boldsymbol{x}} = \boldsymbol{a} \tag{17.21}$$

which can be considered as a new system input. Equation 17.21 has the character of a linearized and decoupled system of second order. By choosing a control law of the form

$$\boldsymbol{a} = \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_{Px}(\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{K}_{Dx}(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) \tag{17.22}$$

the closed-loop control is realized in Cartesian space by use of the current pose of the end effector and therefore considers end effector dynamics and wire forces

$$\boldsymbol{F}_v = \boldsymbol{M}_{eq}(\boldsymbol{x}_d)(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_{Px}\boldsymbol{e}_x + \boldsymbol{K}_{Dx}\dot{\boldsymbol{e}}_x) + \boldsymbol{N}(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d). \tag{17.23}$$

In order to save processing power, the inertia matrix $\boldsymbol{M}_{eq}$ and the vector $\boldsymbol{N}$ are calculated by use of the desired states $\boldsymbol{x}_d$ and $\dot{\boldsymbol{x}}_d$. The resulting virtual force vector $\boldsymbol{F}_v$ has to be mapped onto the motor torques. This is done by use of the active set method according to (17.15). While the previous control approach considers the system dynamics, joint control

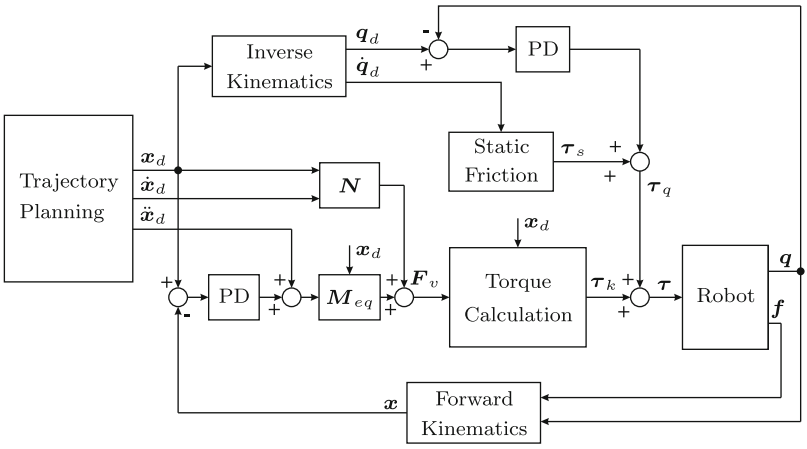



**Fig. 17.3** Control scheme

$$\boldsymbol{\tau}_q = \boldsymbol{K}_{Pq}(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_{Dq}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_s(\dot{\boldsymbol{q}}_d) \tag{17.24}$$

is needed to compensate for the known static friction forces working on the actuators. Now (Fig. 17.3)

$$\boldsymbol{\tau} = \boldsymbol{\tau}_q + \boldsymbol{\tau}_k \tag{17.25}$$

is used for computing the motor torque command, with $\boldsymbol{\tau}_k = f(\boldsymbol{F}_v, \boldsymbol{A}^T)$. The end effector with an assumed mass of 100 kg has to perform oscillating movements along the z-axis with an amplitude of 0.1m at 0.25Hz and around the local x-axis with an amplitude of 30° at 2.5Hz. Within Fig. 17.4 the simulated time history of the pose error is shown. Considering the dimensions of the wind tunnel test bed with $L = 5.2$m, $B = 3.8$m, $H = 2.2$m and the assumed length of the wires with $\boldsymbol{l} = [1.9 \quad 1.9 \quad 1.9 \quad 1.6 \quad 1.6 \quad 1.6 \quad 1.6]^T$m a quite remarkable control quality has been achieved. Figure 17.4 represents the time history of the wire forces which stay below a desired maximum value of 2,000N (Fig. 17.5).

## 17.4 Development of Storage and Retrieval Machine for High Racks

Intralogistics systems are a rapidly growing market having a major influence onto production costs of industrial goods. Industrial goods must be stored, reorganized and delivered—whether for internal production purposes or external distribution. Nowadays high racks and storage retrieval machines are widely used where goods are usually stored in standardized containers.

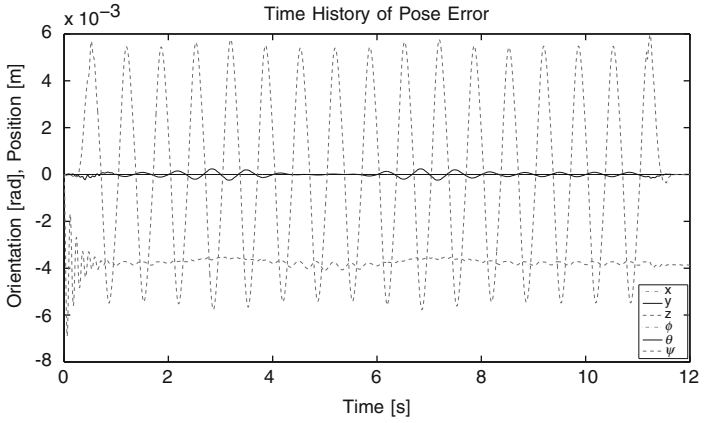



**Fig. 17.4** Time history of pose error

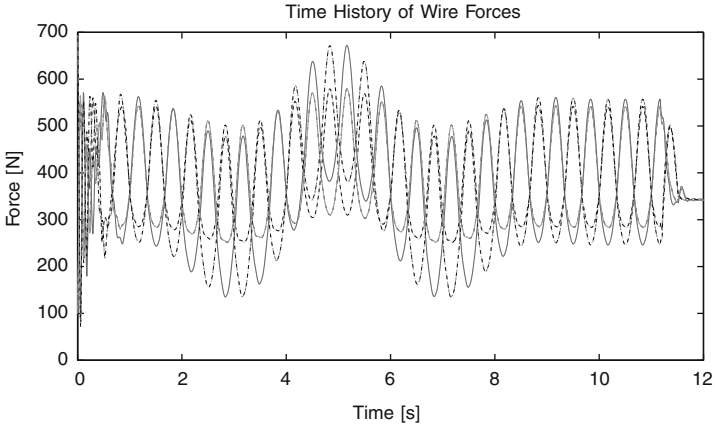



**Fig. 17.5** Time history of wire forces

High racks are very effective in storing goods on small footprints. Additionally they allow the application of fully automated storage and retrieval machines. But the conventional setup shows one drawback: The higher the racks are, the heavier the storage and retrieval machine becomes due to stiffness requirements. While the containers or goods to be moved are often very lightweight (in case of small-parts warehouse, below 50kg), the storage and retrieval machine may weight up to 2t.

The reason for this poor mass ratio between cargo and transport system is mainly related to the serial structure of the storage and retrieval machine: In a kinematical sense, the storage and retrieval machine consists of two prismatic joints connected by links—namely, a skid on a rail carrying the lifter mechanism.

Usually a parallel system has the potential to massively reduce the moved masses, but in the application at hand, a conventional parallel kinematic approach

usually would suffer from a very limited workspace. But since Landsberger and Sheridan [14] presented the concept of a wire-based parallel kinematic machine, this drawback can be efficiently avoided. In the project presented here, a wire-based storage and retrieval machine is developed: The German Federal Ministry of Education and Research started the largest European logistics research initiative ever—the "EffizienzCluster LogistikRuhr" framework. Within this framework, a sub-project called "Storage Retrieval Machine based on the Stewart-Gough-Platform" will develop a storage retrieval machine using a parallel wire robot. The basic idea of the project is to suspend only the lightweight transport platform with a gripper by eight pre-stressed tendons in a parallel configuration. There is a number of—mostly well-known—advantages expected due to this concept:

- A wire robot can be easily adapted to different load ranges and goods to be carried.
- A wire robot is intrinsically lightweight. This reduces the energy consumption or allows faster movements, respectively. Explicitly, the so-called single cycle time (travel time), defined as the time to move the platform from the input/output point to a storage position and back, and the double cycle time, defined as the time to move the platform from the input/output point to a storage position, to another storage position and back to the input/output point, are crucial. The serial structure of the conventional storage retrieval system requires racks which are usually very long compared to their height. Wire robots take advantage from the opposite ratio. This allows very large storage capacities on the same base footprint, allowing for extremely high storage capacities.
- The mechanical structure of the wire robot system is simple and can be realized using cheap components from stock. On the other hand, a complex control system including computerized winches must be developed.

Presently, there is a number of upcoming application projects for wire robots which also base on these properties. These projects cover a wide application range including e.g. suspension systems for wind tunnels [3, 4, 8, 13, 20], aquatic applications [1], positioning of large-scale telescope receivers [21, 22] and assembly of solar power plants [18].

The idea of using a wire robot for high storage retrieval machines was already presented in the past by Hassan and Khajepour [12]. They propose and optimize a geometrical configuration, calculate the wrench-closure workspace and perform a stiffness analysis. Within the presented project, one of the first steps was the development of a virtual prototype. It allows first estimations of the expected performance and energy consumption. The model uses massless wires with an E-module of 65GPa and a diameter of 5mm, assuming linear elasticity and damping.

### *17.4.1 Power Requirements*

First of all, the dimensioning of the motor winches had to be defined. That was done based upon the desired performance defined in Table 17.1.

**Table 17.1** Drive and power requirements

| Property | Value |
|---|---|
| Maximum jerk | $25\frac{\mathrm{m}}{s^3}$ |
| Maximum acceleration | $5\frac{\mathrm{m}}{s^2}$ |
| Maximum velocity | $6\frac{\mathrm{m}}{s}$ |
| Mass of platform including load | 100kg |

A test trajectory was specified covering a variety of typical platform movements. The path is shown in Fig. 17.6a and its trajectory in Fig. 17.6b.

Based on inverse kinematics calculations, the lengths, velocities and accelerations of each wire were derived. The results are shown in Fig. 17.6c. Based upon the given trajectories, the corresponding dynamics were computed which lead to the platform wrench that had to be provided by the wires to perform the desired maneuver. Having the platform wrench at hand, wire force distribution for each point of the trajectory were computed where a minimum wire force $f_{\min} = 100\mathrm{N}$ was specified. The results are shown in Fig. 17.6d.

Having the wire force and the wire velocities at hand, winch rotary speeds (Fig. 17.6e) and torques (Fig. 17.6f) could be evaluated. As shown in Fig. 17.6g, h the drive and power supply requirements according to Table 17.2 could be derived.

## *17.4.2 Optimal Trajectory Generation*

To evaluate the full potential of the wire-driven system, an optimal control approach was chosen to generate optimal trajectories. Optimal trajectories may be defined in the sense of shortest time, least peak power, lowest energy consumptions, or weighted combinations of these. Since the optimal control strategy is based on the dynamics of the wire robot according to (17.8) the optimal trajectory is best adapted for this particular robot.

### 17.4.2.1 Optimal Control Problem

First of all, the problem of optimal control is introduced. The nonlinear state space model of a dynamic system is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{17.26}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ are the state variables, $\boldsymbol{u}(t) \in \mathbb{R}^m$ the control variables, $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \times [0, t_f] \to \mathbb{R}^n$ is continuously differentiable within the time interval $0 \leq t \leq t_f$ and $t, t_f \in \mathbb{R}$. In the problem at hand, a minimum time trajectory is desired,

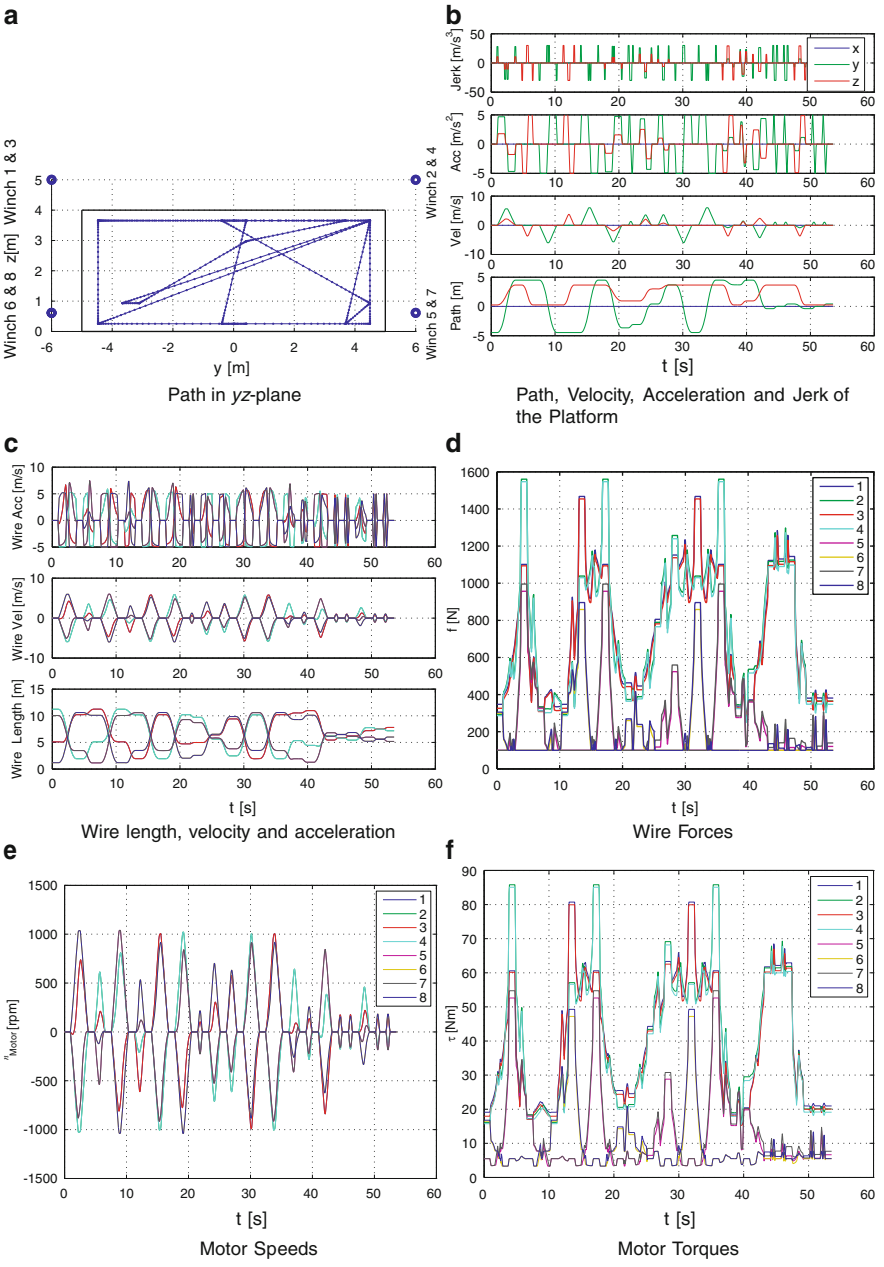



**Fig. 17.6** (continued)

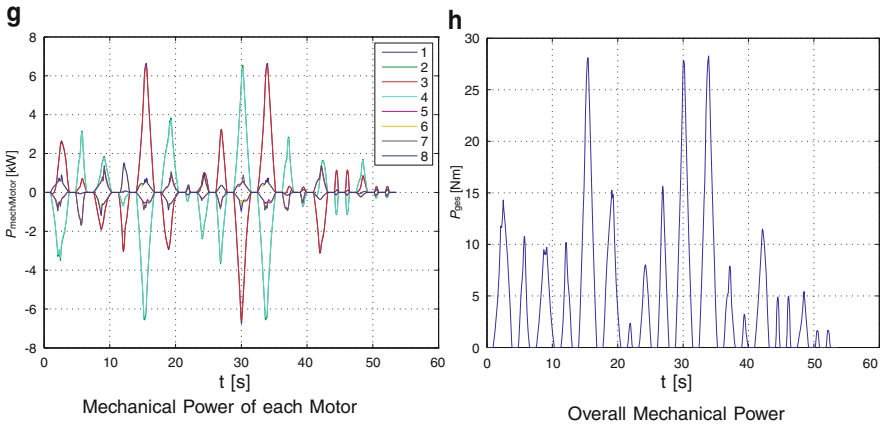



**Fig. 17.6** Simulation results on test trajectory

**Table 17.2** Drive and power requirements

| Property | Symbol | Value |
|---|---|---|
| Maximum speed | $n_{\max}$ | 1,050min$^{-1}$ |
| Maximum torque | $\tau_{\max}$ | 86Nm |
| Maximum power per drive | $p_{\text{mech, mot, max}}$ | 7kW |
| Maximum overall power | $p_{\text{mech, total, max}}$ | 30kW |

whereby the final time $t_f$ is not known a priori. For this reason, a transformation to the normed time interval [0, 1] is necessary. Thereby, $t_f$ becomes part of the variables to be optimized, such that we seek a control function $\boldsymbol{u}(t)$ which minimizes the objective function

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t),t_f} \quad \phi(\boldsymbol{x}(1), t_f) + \int_0^1 t_f \cdot f_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\,\mathrm{d}t \tag{17.27a}$$

$$s.t. \quad \dot{\boldsymbol{x}} = t_f \cdot \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{17.27b}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \qquad \boldsymbol{x}(1) = \boldsymbol{x}_f \tag{17.27c}$$

$$\boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t_f) \leq \boldsymbol{0}, \tag{17.27d}$$

$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t_f) = \boldsymbol{0}, \quad t \in [0, 1] \tag{17.27e}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ and $f_0 : \mathbb{R}^n \times \mathbb{R}^m \times [0, 1] \to \mathbb{R}$ are continuously differentiable. Moreover, (17.27c) describes the initial and final constraints of the trajectory. By (17.27d) and (17.27e) we can consider equality as well as inequality constraints of the state and control variables as well as the final time.

Due to the time-continuous state and control variables the optimal control problem according to (17.27) is in principle an infinite dimensional optimization problem and cannot be solved analytically. For this reason, we have to resort to numerical solution methods which can be divided into direct and indirect methods. In the present paper, a direct method is applied which is based on the finite discretization of the time interval [0, 1] where the state and control variables are approximated according to [11]. Detailed information on optimal control and its numerical solution can also be found in [9, 19].

#### 17.4.2.2 Trajectory Constraints

Before setting up the optimal control problem for this specific robot, several technical requirements need to be provided for the optimal trajectory.

Firstly, the platform orientation has to be parallel to the floor and rack throughout the trajectory such that $\boldsymbol{\Omega}(t), \dot{\boldsymbol{\Omega}}(t), \ddot{\boldsymbol{\Omega}}(t), \dddot{\boldsymbol{\Omega}}(t) = \mathbf{0}$. For this reason (17.8) becomes a differential-algebraic system of equations according to

$$m_p \ddot{\boldsymbol{r}} = \boldsymbol{A}_t^{\mathrm{T}} \boldsymbol{f} + \boldsymbol{f}_E, \qquad \mathbf{0} = \boldsymbol{A}_r^{\mathrm{T}} \boldsymbol{f} + \boldsymbol{\tau}_E, \tag{17.28a}$$

where $\boldsymbol{A}^{\mathrm{T}} = [\boldsymbol{A}_t^{\mathrm{T}}, \boldsymbol{A}_r^{\mathrm{T}}]^{\mathrm{T}}$ is separated into a "translational" and "rotational" structure matrix.

It is obvious that at start and final position the platform velocity and acceleration are zero. Since the platform accelerations are explicitly given in the boundary conditions, they must be integrated into the state variables. In addition, we may consider minimum force changes during the optimization process. For this purpose the platform jerk as well as the force changes are defined to be the control variables. Thus, we obtain the system of DAEs

$$[\dot{\boldsymbol{x}}_1, \dot{\boldsymbol{x}}_2, \dot{\boldsymbol{x}}_3, \dot{\boldsymbol{x}}_4]^{\mathrm{T}} = [\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{u}_1, \boldsymbol{u}_2]^{\mathrm{T}} \tag{17.29a}$$

$$\mathbf{0} = \boldsymbol{A}_r^{\mathrm{T}} \boldsymbol{x}_4 + \boldsymbol{f}_e, \tag{17.29b}$$

$$\mathbf{0} = \tfrac{1}{m}\left(\boldsymbol{A}_t^{\mathrm{T}} \boldsymbol{x}_4 + \boldsymbol{\tau}_e\right) - \boldsymbol{x}_3, \tag{17.29c}$$

where $\boldsymbol{x}_1 = \boldsymbol{r}$, $\boldsymbol{x}_2 = \dot{\boldsymbol{r}}$, $\boldsymbol{x}_3 = \ddot{\boldsymbol{r}}$ and $\boldsymbol{x}_4 = \boldsymbol{f}$. Note the additional algebraic (17.29c) which essentially describes NEWTON'S equation. We obtain the following boundary conditions

$$\boldsymbol{x}_1(0) = \boldsymbol{r}_0, \quad \boldsymbol{x}_2(0) = \mathbf{0}, \quad \boldsymbol{x}_3(0) = \mathbf{0}, \quad \boldsymbol{x}_4(0) = \boldsymbol{f}_0, \tag{17.30a}$$

$$\boldsymbol{x}_1(1) = \boldsymbol{r}_f, \quad \boldsymbol{x}_2(1) = \mathbf{0}, \quad \boldsymbol{x}_3(1) = \mathbf{0}, \quad \boldsymbol{x}_4(1) = \boldsymbol{f}_f, \tag{17.30b}$$

whereby the wire force distributions $\boldsymbol{f}_0$ and $\boldsymbol{f}_f$ are determined from the static wire force calculation in (17.9) by setting the time derivatives of $\boldsymbol{w}$ to zero and solving for $\boldsymbol{f}$. Moreover, we have some additional constraints which have to be valid for $t \in [0, 1]$. Thereby, the platform is not allowed to move beyond the length and height of the rack while considering its maximum velocity, acceleration and jerk. Furthermore, the limited wire forces are to be considered as well. Thus, we have the linear

$$\boldsymbol{r}_{\min} \leq \boldsymbol{x}_1(t) \leq \boldsymbol{r}_{\max}, \qquad \boldsymbol{f}_{\min} \leq \boldsymbol{x}_4(t) \leq \boldsymbol{f}_{\max} \tag{17.31}$$

and nonlinear

$$\boldsymbol{x}_2^2(t) \leq v_{\max}^2, \quad \boldsymbol{x}_3^2(t) \leq a_{\max}^2, \quad \boldsymbol{u}_1^2(t) \leq r_{\max}^2 \tag{17.32}$$

constraint inequalities for this optimal control problem.

#### 17.4.2.3 Objectives

The following objectives are useful to be considered. Firstly, it is reasonable to decrease the final time $t_f$ as much as possible, such that simply

$$\phi(t_f) = t_f. \tag{17.33}$$

Moreover, we would like to keep the driving power as low as possible:

$$f_{0,1}(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \boldsymbol{x}_4(t)) = \left(\boldsymbol{x}_4^{\mathrm{T}}(t)\, \dot{\boldsymbol{l}}(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))\right)^2, \tag{17.34}$$

where $\dot{\boldsymbol{l}}(\boldsymbol{x}_1(t), \boldsymbol{x}_2(t))$ is the wire length change as described in [7]. It may also be desirable to minimize the wire force distributions in order to consume minimum energy according to $f_{0,2}(\boldsymbol{x}_4(t)) = \boldsymbol{x}_4^2(t)$. Further objective criteria such as minimum changes of the wire force distribution and minimum platform jerk can be considered. One can also apply a weighted linear combination of the different criteria to an overall objective according to:

$$\min_{\boldsymbol{x}(t), \boldsymbol{u}(t), t_f} \alpha_0\, \phi(\boldsymbol{x}(1), t_f) + \sum_{i=1}^{m} \alpha_i \int_0^1 t_f \cdot f_{0,i}\,(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\, \mathrm{d}t, \tag{17.35}$$

where $\alpha_i$, $0 \leq i \leq m$ are the corresponding weighting factors.

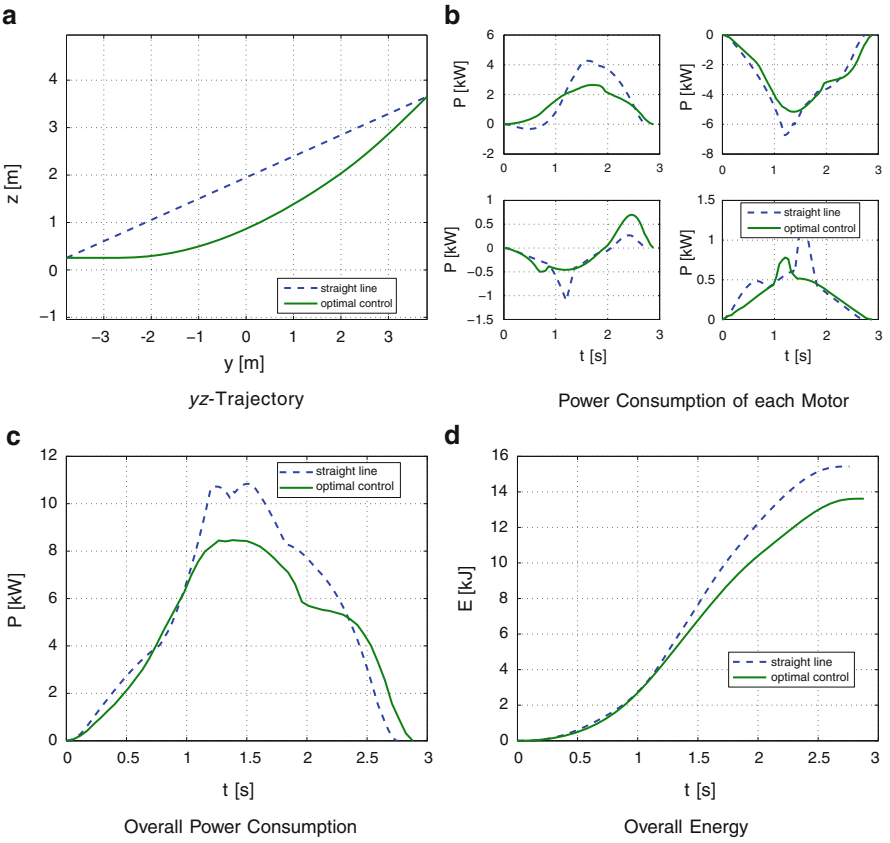



**Fig. 17.7** Diagonal trajectory

### 17.4.2.4 Simulation Results

The optimal trajectories discussed in the following are compared to a reference trajectory which is a straight line from start to final point under the same boundary constraints, maximum platform velocity, acceleration and jerk as well as force limitations and zero platform orientation. This straight line trajectory is generated such, that the platform moves as fast as possible from start to final position.[1]

The following simulation results are based on the desired parameters $v_{\max} = 6\ \frac{\mathrm{m}}{\mathrm{s}}$, $a_{\max} = 5\ \frac{\mathrm{m}}{\mathrm{s}^2}$, $r_{\max} = 30\ \frac{\mathrm{m}}{\mathrm{s}^3}$, $f_{\min} = 100$ N and $f_{\max} = 5{,}000$ N. The weighting factors of the objective criteria are set to $\alpha_0 = 10^3$ and $\alpha_1 = 10^{-5}$ for the minimum time and consumed power.

Figure 17.7 shows a PTP trajectory from the lower left to the upper right corner of the rack. While the simulation time of the straight line trajectory needs about

[1] There exists an analytical solution to this trajectory which is also easy to implement.

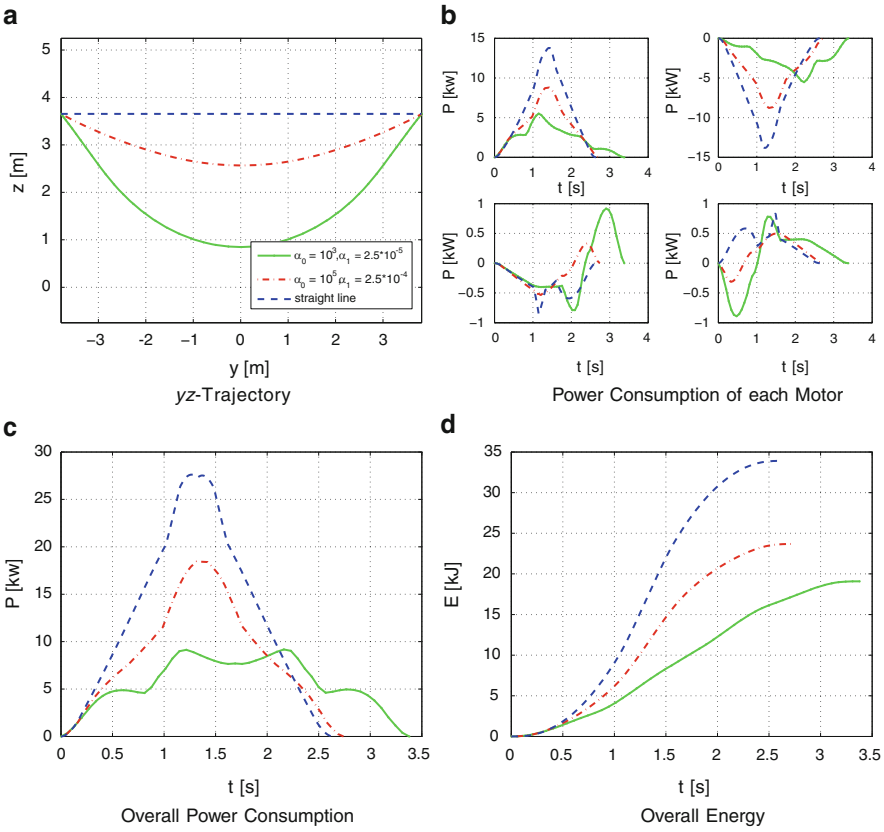



**Fig. 17.8** Horizontal trajectory

2.75 s, the optimal trajectory requires about 2.798 s which is approximately 4.71 % slower than the former. However, the peak power consumption of each motor can generally be reduced. For instance, the second motor needs a peak power of about 6.73 kW for the straight line trajectory, whereas for the optimal trajectory a peak power of about 5.16 kW is required. This equals a peak power reduction of about 23.26 % and consequently means that smaller motors can be applied for this particular robot. Accordingly, the overall peak power consumption can be reduced by about 21.91 %. Moreover, the energy needed to perform this trajectory can be decreased by about 11.73 %.

The following simulation results in Fig. 17.8 show a horizontal trajectory from the upper left to upper right corner of the rack.

The solid lines depict the trajectory and its power and energy consumption for the same weighting factors used above. Concerning this trajectory one can see a time increase of about 28.46 %, however a power and energy saving of about 66.77 % and 43.72 % compared to the straight line trajectory. This time increase may be too much despite the improved power and energy consumption. For this

reason a higher weight of the time factor can decrease the trajectory time to only 4.1941 % as can be seen by the dash-dotted lines. Of course, power as well as energy consumption are then reduced to 33.22 % and 30.17 %, respectively.

**Acknowledgement** This work is based on the results of the projects SEGESTA (SEILGETRIEBENE STEWART-PLATTFORMEN IN THEORIE UND ANWENDUNG), supported by the Germany Research Counsil DFG under HI 370/18, and ARTIST (ARBEITS-RAUMSYNTHESE SEILGETRIEBENER PARALLELKINEMATIK-STRUKTUREN), supported by the Germany Research Counsil DFG under HI370/24-1 and SCHR1176/1-2. The authors would like to thank the German Federal Ministry of Education and Research which supports the project "Regalbediengerät auf Basis der Stewart-Gough-Plattform" within the "EffizienzCluster LogistikRuhr" framework.

## References

1. Borgstrom PH, Jordan BL, Borgstrom BJ, Stealey MJ, Sukhatme GS, Batalin MA, Kaiser WJ (2009) NIMS-PL: a cable-driven robot with self-calibration capabilities. Trans Robot 25(5):1005–1015. doi:http://dx.doi.org/10.1109/TRO.2009.2024792
2. Brown GW (1987) Suspension system for supporting and conveying equipment, such as a camera. US Patent 4,710,819, December 1987
3. Bruckmann T (2010) Auslegung und Betrieb redundanter paralleler Seilroboter. Ph.D. dissertation, Universität Duisburg-Essen, Duisburg, Germany. http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=23303
4. Bruckmann T, Hiller M, Schramm D (2010) An active suspension system for simulation of ship maneuvers in wind tunnels. In: Pisla D, Ceccarelli M, Husty M, Corves B (eds) New trends in mechanism science, Mechanisms and machine science, vol 5, Springer, Dordrecht. ISBN: 978-90-481-9688-3
5. Bruckmann T, Mikelsons L, Brandt T, Hiller M, Schramm D (2008) Wire robots, part I—kinematics, analysis & design. In: Lazinica A (ed) Parallel manipulators—new developments. ARS Robotic Books, I-Tech Education and Publishing, Vienna, Austria, pp 109–132. ISBN 978-3-902613-20-2
6. Bruckmann T, Mikelsons L, Brandt T, Hiller M, Schramm D (2008) Wire robots, part II—dynamics, control & application. In: Lazinica A (ed) Parallel manipulators—new developments. ARS Robotic Books, I-Tech Education and Publishing, Vienna, Austria, pp 133–153. ISBN 978-3-902613-20-2
7. Bruckmann T, Mikelsons L, Hiller M (2008) A design-to-task approach for wire robots. In: Kecskeméthy A (ed) Proceedings of conference on interdisciplinary applications of kinematics 2008, Lima, Peru, 9–11 Jan 2008
8. Bruckmann T, Mikelsons L, Pott A, Abdel-Maksoud M, Brandt T, Schramm D (2009) A novel tensed mechanism for simulation of maneuvers in wind tunnels. In: Proceedings of the ASME 2009 international design engineering technical conferences & computers and information in engineering conference, ASME International, San Diego, CA
9. Dickmanns ED, Well KH (1974) Approximate solution of optimal control problems using third order hermite polynomial functions. In: Optimization techniques'74, pp 158–166
10. Fang S (2005) Design, modeling and motion control of tendon-based parallel manipulators. Ph.D. dissertation, Gerhard-Mercator-University, Duisburg, Germany
11. Geiger BR, Horn JF, DeLullo AM, Long LN (2006) Optimal path planning of uavs using direct collocation with nonlinear programming. In: Proceedings of AIAA GNC conference, Keystone, 21–24 Aug 2006
12. Hassan M, Khajepour A (2009) Analysis of a large-workspace cable-actuated manipulator for warehousing applications. In: Proceedings of the ASME 2009 international design engineering

technical conferences & computers and information in engineering conference. ASME International, San Diego, CA
13. Lafourcade P, Llibre M, Reboulet C (2002) Design of a parallel wire-driven manipulator for wind tunnels. In: Gosselin CM, Ebert-Uphoff I (eds) Workshop on fundamental issues and future research directions for parallel mechanisms and manipulators, Quebec City, 3–4 Oct 2002
14. Landsberger S, Sheridan T (1985) A new design for parallel link manipulator. In: International conference on cybernetics and society, Tucson, Arizona, pp 812–814
15. Merlet JP (2008) Kinematics of the wire-driven parallel robot marionet using linear actuators. In: IEEE international conference on robotics and automation, Pasadena, pp 3857–3862
16. Ming A, Higuchi T (1994) Study on multiple degree of freedom positioning mechanisms using wires, part 1—concept, design and control. Int J Jpn Soc Precis Eng 28:131–138
17. Pott A, Bruckmann T, Mikelsons L (2009) Closed-form force distribution for parallel wire robots. In: Proceedings of computational kinematics 2009, Springer, Berlin/Heidelberg, pp 25–34
18. Pott A, Meyer C, Verl A (2010) Large-scale assembly of solar power plants with parallel cable robots. In: ISR/ROBOTIK 2010, Munich, Germany
19. Stryk O (1993) Numerical solution of optimal control problems by direct collocation
20. Sturm C, Bruckmann T, Schramm D, Hiller M (2011) Optimization of the wire length for a skid actuated wire based parallel robot. In: Proceedings of the 13th World Congress in Mechanism and Machine Science, (IFToMM2011), Guanajuato, México, 19–25 June, 2011
21. Taghirad H, Nahon M (2007) Forward kinematics of a macro–micro parallel manipulator. In: Proceedings of the 2007 IEEE/ASME international conference on advanced intelligent mechatronics (AIM2007), Zurich, Switzerland
22. Taghirad H, Nahon M (2007) Jacobian analysis of a macro–micro parallel manipulator. In: Proceedings of the 2007 IEEE/ASME international conference on advanced intelligent mechatronics (AIM2007), Zurich, Switzerland
23. Vafaei A, Aref MM, Taghirad HD (2010) Integrated controller for an over-constrained cable driven parallel manipulator. In: IEEE international conference on intelligent robots and systems, Anchorage, AK, USA, pp 650–655
24. Yaqing Z, Qi L, Xiongwei L (2007) Initial test of a wire-driven parallel suspension system for low speed wind tunnels. In: Proceedings on 12th IFToMM world congress, Besançon, France, 18–21 June 2007

# Index



H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2, 