178

Budapest University of Technology and Economics

Computed Torque Control and Utilization of Parametric Excitation for Underactuated Dynamical Systems

by

Ambrus Zelei

A thesis submitted in partial fulfillment for the degree of Doctorate of Philosophy in Mechanical Engineering

Supervisor: Dr. Gábor Stépán

Budapest, February 27, 2015

Declaration of Authorship

I, Ambrus Zelei, hereby declare that this master thesis titled, 'Computed Torque Control and Utilization of Parametric Excitation for Underactuated Dynamical Systems' and the work presented in it are my own. I confirm that all relevant resources are marked I have used while working on the thesis.

Signed:

Date:

Abstract

Dynamical systems with less independent control input than degrees of freedom are called underactuated systems. They form a special group of robotic systems, because they are more energy efficient and agile compared to the classical industrial robots having heavy mechanical structure and robust actuators at each joint. Cranes are typical underactuated systems because there is no direct actuation on the swinging payload.

The present work is motivated by a newly designed domestic robot called Acroboter, which moves on a specially designed ceiling and the working unit of the robot hangs down and operates in the 3D workspace like a crane. Since the robot has a complex multibody structure, the dynamic modeling requires a special approach, where non-minimum set of redundant coordinates describes the system instead of the classical minimum set of generalized coordinates. Geometric constraints are introduced to represent the relations among the redundant coordinates. The corresponding dynamical model is a system of differential algebraic equations. The present work addresses the development of model based motion control algorithms for underactuated multibody systems, in general.

As an application of the results, the proposed control algorithms are applied for varying topology systems, like fully actuated systems in the presence of actuator saturation. Actuator saturation is a relevant nonlinearity, which is treated here as a decrement in the number of independent control inputs. Another group of varying topology underactuated systems in focus belong to the limbless locomotion.

One of the most intricate problems is when certain tasks are prescribed for the passive DoF of an underactuated system. By augmenting the actuator forces with some periodic excitation for the active DoF, the tasks could be approached even for the passive DoF. Since this periodic excitation at the actuators usually presents some time-periodic parameters in the equations of motion, this kind of forcing is called parametric excitation in classical mechanics. In this sense, parametric excitation could succesfully be used for the control of certain underactuated systems. Case studies of stabilization of water vessels and the control of pendulum-like robots via parametric excitation are presented.

Finally, the motion control of the Acroboter is accomplished, which is partially based on closed form formulae derived from simplified pendulum-like models of the robot. The simplified control appoaches are combined with the general methods derived in the first part of the dissertation. The control approaches are tested and applied in laboratory experiments for the Acroboter prototype.

Keywords: underactuated systems, kinematic redundancy, computed torque control, multibody systems, parametric excitation, crane-like robots

Összefoglaló

Alulaktuáltnak nevezzük azokat a dinamikai rendszereket, amelyek kevesebb független szabályozási bemenettel rendelkeznek, mint amennyi szabadsági fokuk van. A robotok egy speciális csoportját alkotják, mert energiahatékonyabbak és fürgébbek a leginkább elterjedt ipari robotoknál, amelyek nagy tömegű vázszerkezettel és robusztus hajtómotorokkal rendelkeznek minden egyes csuklóban. A daruk tipikus alulaktuált rendszerek, mivel nincs közvetlen ráhatás a lengő teherre.

A jelen munkát egy újonnan tervezett háztartási robot, az Acroboter motiválta, amely egy speciálisan kialakított mennyezeten mozog, a robot változtatható hosszúságú kábelekkel felfüggesztett munkavégző egysége pedig a darukhoz hasonlóan végzi feladatát a 3 dimenziós munkatérben. A robot összetett mechanikai szerkezettel rendelkezik, ezért dinamikai modellezése egy speciális, a többtestdinamika területéről ismert megközelítést igényel, miszerint nem minimális számú redundáns koordinátával írjuk le a rendszert a klasszikus általános koordinátás leírásmód helyett. A redundáns koordináták között geometriai kényszeregyenletek teremtik meg a kapcsolatot, emiatt a rendszer dinamikai modellje differenciál algebrai egyenletek formájában adható meg. A jelen munka alulaktuált többtest-dinamikai rendszerekre általánosan alkalmazható, modell alapú mozgásszabályozási algoritmusok kidolgozását célozza.

Az eredmények egy lehetséges alkalmazásaként változó topológiájú rendszerekre általánosítom a kidolgozott algoritmusokat, többek között olyan teljes aktuáltságú robotok szabályozására, amelyeknél a beavatkozók telítődését figyelembe kell venni. A hajtómotorok telítődése egy lényeges nemlinearitás, amit jelen esetben a független szabályozási bemenetek számának csökkenéseként kezelek. A változó topológiájú rendszerek egy másik esete, amelyet ugyancsak vizsgálok, a láb és kerék nélküli helyváltoztatásra képes szerkezetek csoportja.

Alulaktuált rendszerekre a leginkább összetett problémák akkor adódnak, ha bizonyos feladatok a passzív szabadsági fokokra vannak előírva. Az aktív szabadsági fokokra ható szabályozás periodikus gerjesztéssel történő kiegészítésével a passzív szabadsági fokokra előírt feladatok is teljesíthetőek az aktív szabadsági fokokra előírt feladatok betartása mellett. Mivel az alkalmazott periodikus gerjesztés általában valamilyen periodikus paraméter jelenlétét okozza a mozgásegyenletben, ezt a fajta gerjesztést paraméteres gerjesztésnek nevezik a klasszikus mechanikában. Ilyen értelemben a paraméteres gerjesztés alkalmazható bizonyos alulaktuált rendszerek szabályozására. Vízi járművek és inga-szerű robotok paraméteres gerjesztéssel történő szabályozására esettanulmányokat dolgozok ki.

Végül az Acroboter mozgásszabályozását mutatom be, amely részben a robot egyszerűsített, inga-szerű modelljeinek segítségével nyert zárt alakú formulákon alapul. Az egyszerűsített szabályozási algoritmusokat a dolgozat elején bemutatott általános módszerekkel kombinálva az Acroboter prototípuson sikeres laboratóriumi teszteket hajtottunk végre.

Kulcsszavak: alulaktuált dinamikai rendszerek, kinematikai redundancia, kiszámított nyomatékok módszere, többtest-dinamikai rendszerek, paraméteres gerjesztés, daru szerű robotok

Acknowledgements

Over the past years I have received support and encouragement from a great number of individuals. I would like to express my deepest gratitude to my supervisor Gábor Stépán for the patience and the huge effort during the process of writing this dissertation and for the guidance during the past ten years starting from my undergraduate studies. I would never have been able to finish my dissertation without László L Kovács and László Bencsik. Thanks for the common work and the lot of useful and inspiring discussions. Special thanks to Zoltán Juhász for the common work in the field of worm-like locomotion systems. I express my special thanks to László Kollár and Tamás Insperger for the thorough review of the manuscript. I would like to thank the Acroboter team for making possible to test the theoretical results on a real robotic system. Thanks to my colleagues for the good atmosphere at the department.

I would like to express my thanks to my parents and my sister to give continuous support during my PhD studies especially in the final years. Special thanks to my mother for reading through and correct the manuscript. I also express my thanks to Sára Kukor for the support and the encouragement at the final and hardest period.

This work was supported by the HAS-BME Research Group on Dynamics of Machines and Vehicles under Grant No. MTA-NSF/103 and the Hungarian National Science Foundation (Grant No. OTKA K101714) and the ACROBOTER Project (no.: IST-2006-045530) and the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

Patents pending:

Appl. No.: HU-P0900466. Appl. date: July 28, 2009. Title: "Payload suspension system".
Appl. No.: HU-P0900467. Appl. date: July 28, 2009. Title: "Suspended payload platform thrusted by fluid mass flow generators".

Contents

D	eclar	ation of Authorship	iii
A	bstra	\mathbf{ct}	iv
Ö	sszefe	oglaló	vi
A	cknov	wledgements	viii
Li	st of	Figures	xv
Li	st of	Tables	xvii
ѕу	mbo	ls	xix
A	bbre	viations	xxi
1	Intr	oduction	1
2	Cor 2.1 2.2 2.3 2.4 2.5	nputed torque control methods in the literature for underactuated systems Computed torque control method in general Formulation of underactuated systems dynamics Relative degree for feedback linearization of SISO systems Partial feedback linearization of collocated and non-collocated serial robots Computed desired computed torque control	3 3 5 7 9 11
3	Cor 3.1 3.2	Imputed torque control of systems with redundant coordinates Mathematical background	13 13 13 15 15 16 18 19 20 21 22 24 25 25 27 28

	3.3	Extension of Lagrange multipliers method for CTC of underactuated systems	29
		3.3.1 Two-steps approach	29
		3.3.2 Direct approach	30
	3.4	Demonstrative comparison of CTC methods	30
		3.4.1 Mechanical model for case study	31
		3.4.2 Definition of tasks and corresponding relative degrees	33
		3.4.3 Comparison of control strategies by numerical simulations	36
		3 4 4 Concluding remarks	37
	3.5	New results	38
	0.0		00
4	App	lications for underactuated systems of varying topology	41
	4.1	Worm-like locomotions	41
		4.1.1 Limbless locomotion	41
		4.1.2 Mechanical model of a worm-like locomotion	42
		4.1.2.1 Parametrization	43
		4.1.2.2 Geometric constraints	43
		4.1.2.3 Stiffness and damping added to the worm body	43
		4.1.2.4 Actuation of the worm model	45
		4.1.2.5 Contact with the ground	45
		4.1.3 Numerical simulation	46
		4.1.4 Optimization	47
		4.1.5 Concluding remarks	 47
	4.2	Underactuation induced by actuator saturation	 48
		4.2.1 Combined fully actuated and underactuated algorithm for handling saturation	49
		4.2.2 Dimension reduction of servo-constraint in specific cases	50
		4.2.2 Dimension reduction of servo-constraint in specific cases	51
		4.2.5 Dimension reduction of servo-constraint in general cases	52
		4.2.4 Case study for all first manipulator $\dots \dots \dots$	52
		4.2.4.1 Specific case	55
		4.2.4.2 General case	00 55
	4.0	4.2.5 Concluding remarks	55
	4.3	New results	50
5	Use	of parametric excitation for underactuated systems	57
	5.1	Pendulum-like robots	57
		5.1.1 Task definition	58
		5.1.2 Classical servo-constraint	59
		5.1.3 Parametric path	59
	5.2	Inverted pendulum-like systems	60
		5.2.1 Stability of row-vessels	60
		5.2.2 Background of parametrically excited systems	61
		5.2.3 Stability of the parametrically excited floating body	62
		5.2.4 Verification by numerical simulation and experiment	65
	5.3	New results	67
	~		
6	Dev	elopement of the Acroboter platform	69
	6.1	Introduction to dynamics of crane-like systems	69
		6.1.1 Lagrangian generalized coordinates and singularities	71
		6.1.2 Cartesian descriptor coordinates	71
	6.2	Structure of Acroboter	72

6.3	Analyt	cical inverse dynamics of pendulum-like models	74
	6.3.1	Single pendulum model	74
	6.3.2	Double pendulum model	76
	6.3.3	Multiple pendulum model	78
	6.3.4	Experimental results	78
6.4	Inverse	e dynamics of planar Acroboter	79
	6.4.1	Analytical approach	80
	6.4.2	Comparison of analytical CTC and backward Euler based CTC	81
6.5	Inverse	e dynamics of spatial Acroboter	85
6.6	Genera	alization for cases of kinematic redundancy	89
	6.6.1	Redundancy resolution	89
	6.6.2	Notion of redundancy in case of underactuated robots	91
	6.6.3	Computed torque method for dynamically redundant systems	91
	6.6.4	Case study for the planar Acroboter	93
6.7	New re	esults	95
Bibliog	raphy		97

List of Figures

1.1	Real life examples for underactuated dynamical systems
$2.1 \\ 2.2$	The position control of a linear 1 DoF system
3.1	Simplified structure (a) and free-body-diagrams (b) of the crane-like robot platform Acroboter
$\begin{array}{c} 3.2\\ 3.3\end{array}$	The experimetal tool (a), the mechanical model (b)26Mechanical models applied for the parameter identification (a,b) and the mechanical26model using natural coordinates27
3.4	Experimental results
3.5	Underactuated and constrained mechanical model for CTC case study 31
3.6	A simulation result of type "A" (succesful control): non-collocated case, relative coor-
3.7	A simulation result of type "B" (unsuccesful control): non-collocated case, absolute coordinates, two-steps MLM
4.1	A simple model for limbless locomotion
4.2	Locomotion techniques of limbless animals: worm's peristaltic motion (a), caterpillar's
19	Mechanical model
4.0	The angle of neighboring rode, and the force from stiffness and demning
4.4	Worm model performing locomotion
4.6	Torsional spring and damper (left) offset of the relaxed angle (right)
4.7	Free body diagram of a lumped mass, and forces acting on the ground
4.8	Constraint violation
4.9	Actuator saturation: nonlinear connection between commanded and real control input 49
4.10	Mechanical model of the studied RR manipulator
4.11	The prescribed initial and end configurations in specific (left) and in general (right) cases
4.12	Numerical results for specific cases A, B and C: servo-constraint violations (left col- umn) and control inputs (right column) 54
4.13	Numerical results for case A, B and C
5.1	Acroboter system (a) path of a pendulum-like robot (b)
5.2	Ince-Strutt diagram: stability chart of Mathieu-equation
5.3	Spatial ship motions (a), planar mechanical model of the floating body (b) 65
5.4	Potencial function of a vertically stable case (a), a vertically unstable case, when two stable equilibria arise (b).
5.5	Stability chart of the parametrically excited floating body with the ω^{e} and y_{1}^{e} values taken from real life experience

5.6	Numerical results: amplitude is too small for the stabilization (a), stable case (b), amplitude is too large for the stabilization (c)	67
5.7	.7 Experimental tool with stopped rotors (a) and running rotors (b)	
6.1	tower crane (a), gantry crane (b), overhead crane (c), floating crane (d) and aerial crane (e)	70
6.2	Redundancy of the varied length pendulum (a), simplified model of a crane (b)	70
6.3	Conceptual design of the Acroboter system	72
6.4	Prototype of the Acroboter system	73
6.5	5 Single pendulum (a) and double pendulum (b) models of the Acroboter platform	
6.6	Numerical example for the inverse calculation of the planar single pendulum and double pendulum models: time histories (a), comparison of the stroboscopic motion of the controlled single and double pendulum (b)	78
6.7	Experimental setup using an early SU prototype and a cartesian robot instead of CU (a) experimental results based on the single pendulum model: dashed lines show the	
	desired and continuous ones show the measured values (b)	79
6.8	The planar, natural coordinates based Acroboter model for numerical simulations	80
6.9	The planar, natural coordinates based Acroboter model for numerical simulations	81
6.10	Numerical simulation of the CTC of the planar Acroboter	84
6.11	Sketch of the natural coordinate based spatial model of the Acroboter	85
6.12	The results of the inverse calculation of the climber unit	86
6.13	Simulation results for the CTC of the spatial Acroboter: the sequences of the motion	
	obtained by means of the single pendulum inverse dynamics model are shown in panels	
	a1, a2, a3 and a4; the inverse kinematics of the CU is shown in panels b1, b2, b3 and	
	b4; the obtained motion of the complete Acroboter model is shown in panels c1, c2,	
	c3, and $c4$	88
6.14	Planar model (a) and free body diagrams (b) of the Acroboter	94
6.15	Simulation results: motion of the Acroboter platform	95
6.16	Simulation results: time histories	95

List of Tables

3.1	Benchmark of the control algorithms for absolute coordinates	36
3.2	Benchmark of the control algorithms for relative coordinates	36
4.1	Optimized parameters	47
5.1	Parameters used for stability investigation	65
5.2	Parameters used for stability investigation	67
6.1	Interpretations of redundancy (n - number of descriptor coordinates, m - the number	
	of geometric constraints, l - dimension of task, g - number of independent actuators)	92

Symbols

The list below is not complete, but contains the most important symbols appearing in this work:

t	time	$[\mathbf{s}]$
I	identity matrix	[1]
n	number of descriptor coordinates	[1]
m	number of geometric constraint equations	[1]
l	number of servo-constraint equations	[1]
g	number of independent control inputs	[1]
$ar{\mathbf{q}}$	minimum set (generalized) coordinate vector	[m] or [rad]
\mathbf{q}	non-minimum set (redundant) coordinate vector	[m] or [rad]
\mathbf{q}^{d}	non-minimum set (redundant) desired coordinate vector	[m] or [rad]
\mathbf{q}_{c}	controlled coordinates	[m] or [rad]
\mathbf{q}_{u}	uncontrolled coordinates	[m] or [rad]
\mathbf{S}_{c}	selector matrix of controlled coordinates	[1]
\mathbf{S}_{u}	selector matrix of uncontrolled coordinates	[1]
\mathbf{q}_{a}	active coordinates	[m] or [rad]
\mathbf{q}_{p}	passive coordinates	[m] or [rad]
u	control input	[SI]
v	synthetic control input	[SI]
$\mathbf{M}(\mathbf{q})$	mass matrix	$[kg], [kgm], [kgm^2]$
$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$	vector of inertial and coriolis terms	[N]
$\mathbf{H}(\mathbf{q})$	control input matrix	[SI]
\mathbf{K}_{P}	proportional gain matrix	[SI]
\mathbf{K}_{D}	derivative gain matrix	[SI]
\mathbf{K}_{lpha}	derivative gain matrix	[SI]
\mathbf{K}_{eta}	proportional gain matrix	[SI]
\mathbf{K}_{γ}	gain matrix	[SI]
$\boldsymbol{\varphi}(\mathbf{q},t)$	geometric costraint vector	[SI]
λ	vector of Lagrange multipliers	[SI]
${oldsymbol \sigma}({f q},t)$	servo-costraint vector	[SI]
$\boldsymbol{\psi}(\bar{\mathbf{q}},\mathbf{p},t)$	parametric function for task definition	[SI]
$\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t)$	non-holonomic servo-costraint vector (optimization rule)	[SI]
x,y,z	coordinates in horizontal directions and in vertical	[m]
h	time step	[s]

Abbreviations

$\underline{\mathbf{o}}$ rdinary $\underline{\mathbf{d}}$ ifferential $\underline{\mathbf{e}}$ quation
<u>d</u> ifferential <u>a</u> lgebraic <u>e</u> quation
$\underline{\mathbf{d}}\mathbf{e}\mathbf{g}\mathbf{r}\mathbf{e}\mathbf{e}\mathbf{s} \ \underline{\mathbf{o}}\mathbf{f} \ \underline{\mathbf{f}}\mathbf{r}\mathbf{e}\mathbf{e}\mathbf{d}\mathbf{o}\mathbf{m}$
\underline{c} omputed \underline{t} orque \underline{c} ontrol
$\underline{\mathrm{c}} \mathrm{omputed} \ \underline{\mathrm{d}} \mathrm{esired} \ \underline{\mathrm{c}} \mathrm{omputed} \ \underline{\mathrm{t}} \mathrm{orque} \ \underline{\mathrm{c}} \mathrm{ontrol}$
$\underline{\mathbf{p}}$ roportional- $\underline{\mathbf{d}}$ erivative
$\underline{\mathbf{c}}$ limber $\underline{\mathbf{u}}$ nit
\underline{s} winging \underline{u} nit
$\underline{c}able \underline{c}onnector$

Chapter 1

Introduction

The purpose of the research work presented in this study is the extension of certain control algorithms for underactuated multibody systems. In classical robotics, the number of degrees of freedom is 6 and the number of actuators is also 6. These multibody systems are fully actuated and they can accomplish essential tasks in the 3D space. However, underactuated multibody systems appear in nature and in engineering almost everywhere. Consider, for example, the human grasping, walking and running [1, 2], the fishes' swimming, the birds' flying and the corresponding engineering structures, like robotic hands, passive walkers, boats, under-water and air vehicles [3], cranes [4, 5], see Fig. 1.1b and c. While the control algorithms of these underactuated systems are much more complicated, their mechanical structures provide energy efficient and agile operation. The elasticity of the mechanical parts of a robotic system like light-weight robots can also be handled as an underactuated system [6, 7], see Fig. 1.1a.



FIGURE 1.1. Real life examples for underactuated dynamical systems

The following definition of underactuated mechanical systems is adapted from [8–11]. Consider a general controlled mechanical system, the mathematical model of which is usually given in the form of a second-order ordinary differential equation (ODE):

$$\ddot{\mathbf{q}} = \mathbf{f}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t) + \mathbf{H}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)\mathbf{u}, \qquad (1.1)$$

where $\bar{\mathbf{q}}$ is the vector of the generalized coordinates of minimum number, $\mathbf{f}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ is a vector field that determines the dynamics of the system, including gravitational, spring, damper forces and also centrifugal and Coriolis forces, gyroscopic effects, and so on. $\mathbf{H}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ is the control input matrix and \mathbf{u} is the control input vector, which represent the actuator forces and torques. The only assumption in the general equation of motion (1.1) is that the control input \mathbf{u} appears linearly. The system is fully actuated, if the rank of the input matrix $\mathbf{H}(\mathbf{\bar{q}}, \mathbf{\dot{\bar{q}}}, t)$ equals to the DoF of the system:

$$\operatorname{rank}(\mathbf{H}(\bar{\mathbf{q}}, \bar{\mathbf{q}}, t)) = \dim(\bar{\mathbf{q}}).$$
(1.2)

We speak about underactuated systems if the number of the independent control inputs is lower than the DoF of the system or in other words, the rank of $\mathbf{H}(\mathbf{\bar{q}}, \mathbf{\dot{\bar{q}}}, t)$ is smaller than the dimension of $\mathbf{\bar{q}}$ [8, 9]:

$$\operatorname{rank}(\mathbf{H}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)) < \dim(\bar{\mathbf{q}}).$$
(1.3)

Overactuated systems also exist, when the number of the independent control inputs is larger than the DoF and more than one actuator can be in connection with one DoF, like in the muscular system of humans and animals. The study of overactuated systems is outside of the focus of the present research work.

The present research was motivated by the development of a domestic robot called Acroboter [12] within the European Union 6th Framework Project (IST-2006-045530) coordinated by the Department of Applied Mechanics, Budapest University of Technology and Economics.

The Acroboter hangs down from the ceiling on a suspension cable similarly to cranes (see Fig. 1.1d), and it is able to utilize the pendulum-like motion efficiently. This specially designed domestic robot has 12 DoF and 10 actuators only, which requires the extension of the existing control algorithms. This extension was necessary partly because the robot has some essential singular configurations when minimum set of general coordinates $\bar{\mathbf{q}}$ are used. This problem can be resolved by means of non-minimum set of the appropriate choice of generalized coordinates, we call them descriptor coordinates. Since the prescribed task related to the position and orientation of a rigid body in 3D space, the task is 6 dimensional only, and consequently, the Acroboter is also a kinematically redundant structure. The corresponding dynamical model is a system of differential algebraic equations. The present work addresses the developement of model-based motion control algorithms for underactuated multibody systems, in general.

As an application of the results, the proposed control algorithms are applied for varying topology systems, like fully actuated systems in the presence of actuator saturation. Actuator saturation is a relevant nonlinearity, which is treated here as a decrement in the number of independent control inputs. Another group of varying topology underactuated systems in focus belong to the limbless locomotion.

One of the most intricate problems is when certain tasks are prescribed for the passive DoF of an underactuated system. By augmenting the actuator forces with some periodic excitation for the active DoF, the tasks could be approached even for the passive DoF. Since this periodic excitation at the actuators usually presents some time-periodic parameters in the equations of motion, this kind of forcing is called parametric excitation in classical mechanics. In this sense, parametric excitation could succesfully be used for the control of certain underactuated systems. Case studies of stabilization of water vessels and the control of pendulum-like robots via parametric excitation are presented.

Finally, the motion control of the Acroboter is accomplished, which is partially based on closed form formulae derived from simplified pendulum-like models of the robot. The simplified control appoaches are combined with the general methods derived in the first part of the dissertation. The control approaches are tested and applied in laboratory experiments for the Acroboter prototype.

Chapter 2

Computed torque control methods in the literature for underactuated systems

This Chapter provides a short literature review on the idea of computed torque control (CTC) and feedback control in general, and the control of underactuated robotic systems.

When the model-based approach, CTC is applied, the inverse dynamics calculation of the dynamical system has to be performed real-time. That is, the actuator torques are determined for the desired motion of the robot in every sampling period.

A special group of controlled dynamical systems is formed by the underactuated ones, in which the number of independent control inputs is lower than the degrees of freedom of the system. In these cases, the application of the CTC leads to a differential algebraic equation (DAE) problem [13, 14] because the generalized coordinates of the system as differential variables and the control inputs as algebraic variables are to be calculated. These can be obtained from the equations that are the results of the coupling of inverse dynamical and inverse kinematical calculations, which cannot be decoupled in case of underactuated systems.

2.1 Computed torque control method in general

The class of *computed torque control* (CTC) methods is based on the technique of applying *feedback linearization* to nonlinear systems [15]. The CTC method is commonly used when the given trajectory of the end-effector or the tool centre point of the robot has to be followed with the smallest possible deviation. The CTC method requires the knowledge of an accurate dynamical model of the robotic system, and the inverse kinematics and dynamics calculations are needed in order to determine the control input [16]. Hence CTC is a *model-based control*. The accurate following of a prescribed trajectory is a typical requirement in industrial robotic systems, like welding manipulators [17], surgical systems [18] or in case of domestic robots such as the Acroboter system [12], among many other examples.

One of the most common ways of controlling robot motion is based on a linear control system obtained by feeding back the dynamics of the original nonlinear system. After this so-called *feedback linearization*, an arbitrary motion can be prescribed, which is realizable until the actuators are able to provide the required torques. This method cannot be directly applied in case of the so-called underactuated systems because the number of the independent control inputs is lower than the DoF of the system. For a general overview of this issue let us consider the equation of motion (1.1). In

case of a fully actuated system, the control input can be formulated as:

$$\mathbf{u} = \mathbf{H}^{-1}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t) \left(-\mathbf{f}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t) + \mathbf{v} \right).$$
(2.1)

by inverting the control input matrix $\mathbf{H}(\mathbf{\bar{q}}, \mathbf{\dot{\bar{q}}}, t)$ and introducing the synthetic input \mathbf{v} to be defined later.

Equation (2.1) is called *control law*, which is, in general, the control action that can be specified as some function of the system's state and the time. This is a more general concept than the earlier idea of feedback since the control law can incorporate both the feedback and the feedforward methods of control. In order to design an inner control loop, (2.1) is substituted into the equation of motion (1.1), from which we obtain the following linear system:

$$\ddot{\mathbf{q}} = \mathbf{v} \,. \tag{2.2}$$

This way, a possibly very complicated nonlinear controller design problem is converted into a simple design problem for a linear system, because a linear mapping is established between the new synthetic input \mathbf{v} and the generalized acceleration $\ddot{\mathbf{q}}$. If we calculate the control input \mathbf{u} according to (2.1) and we measure $\bar{\mathbf{q}}$ and $\dot{\bar{\mathbf{q}}}$ exactly then the acceleration of the system can be prescribed arbitrarily via the synthetic control input \mathbf{v} . For the above explained method, vector $\mathbf{f}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ and matrix $\mathbf{H}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ have to be known exactly, as it is usual in model-based control strategies.

An outer-loop control strategy for the resulting linear control system can then be applied. The synthetic control input \mathbf{v} is chosen, for example, as

$$\mathbf{v} = \ddot{\mathbf{q}}^{d} + \mathbf{K}_{D} \left(\dot{\mathbf{q}}^{d} - \dot{\mathbf{q}} \right) + \mathbf{K}_{P} \left(\bar{\mathbf{q}}^{d} - \bar{\mathbf{q}} \right) , \qquad (2.3)$$

with superscript d referring to desired (or nominal) values. The vectors $\bar{\mathbf{q}}$ and $\dot{\bar{\mathbf{q}}}$ are the measured states of the system. Here, $\mathbf{K}_{\rm D}$ and $\mathbf{K}_{\rm P}$ are constant differential and proportional gain matrices. The linear feedback defined by (2.3) is substituted into the control law (2.1), then the resulting control input \mathbf{u} is substituted back into the equation of motion (1.1). The resulting equation

$$\ddot{\mathbf{q}}^{d} - \ddot{\mathbf{q}} + \mathbf{K}_{D} \left(\dot{\mathbf{q}}^{d} - \dot{\mathbf{q}} \right) + \mathbf{K}_{P} \left(\bar{\mathbf{q}}^{d} - \bar{\mathbf{q}} \right) = \mathbf{0}$$
(2.4)

shows that in case of positive definite gain matrices \mathbf{K}_{P} and \mathbf{K}_{D} the convergence of the tracking error $\bar{\mathbf{q}}^{\mathrm{d}} - \bar{\mathbf{q}}$ to zero is guaranteed, or in other words, the system has stable error dynamics [15].

Note that if the model were perfectly accurate then the nominal control input \mathbf{u} could be calculated offline without any feedback ($\mathbf{K}_{\mathrm{P}} = \mathbf{0}$ and $\mathbf{K}_{\mathrm{D}} = \mathbf{0}$) in (2.4). In that case, the desired value of the state of the system could be used in equation (2.1) by substituting $\mathbf{f}(\mathbf{\bar{q}}^{\mathrm{d}}, \mathbf{\dot{q}}^{\mathrm{d}}, t)$ and $\mathbf{H}(\mathbf{\bar{q}}^{\mathrm{d}}, \mathbf{\dot{q}}^{\mathrm{d}}, t)$ and open loop control would be achieved. However, this is not robust against parameter and model uncertanities and noise or external disturbances, thus a *linear feedback controller*, which is also called *linear compensator*, is used with positive definite gain matrices \mathbf{K}_{P} and \mathbf{K}_{D} .

The advantage of the application of above nonlinear feedback controller (2.1) with linear feedback compensator (2.3) is that they react much faster and they are more accurate compared to a pure linear proportional-derivative (PD) feedback contol. The disadvantage is the high computational demand and the resulting longer sampling times.

In the case of a serial, fully actuated robot manipulator, independent control inputs can be associated with each DoF. Thus, the above CTC method can easily be applied for such systems, especially when they are modeled in the classical way using minimum set of generalized coordinates $\bar{\mathbf{q}}$ and equations of motion in ODE form [15, 16]. In contrast, the control problems are more difficult in case of underactuated robot manipulators in general. For example, the above explained CTC method cannot be accomplished in case of underactuated systems at all, since the non-quadratic $\mathbf{H}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ in (1.1) is not invertible in that case.

Still, the CTC method was generalized for underactuated systems by [7]. This was called computed desired computed torque control (CDCTC) method, where the term "desired" refers to the fact that the desired values of a set of uncontrolled coordinates have to be calculated first, and the control inputs are determined only after the calculation of the so-called desired zero dynamics. This method requires the separation of the generalized coordinates into controlled and uncontrolled ones.

Partial feedback linearization can also be used for the control of underactuated systems [19]. The main idea of this method is to substitute the original nonlinear system with a partially equivalent linear system by means of a nonlinear transformation.

These CTC methods for underactuated systems can be further extended for systems modeled by non-minimum set of descriptor coordinates where additional geometric constraint equations are introduced [6, 20]. In Section 3, alternative ways of the extensions of CTC methods are developed and studied in details in cases of complex underactuated multibody systems where the use of nonminimum set of descriptor coordinates has several advantages.

2.2 Formulation of underactuated systems dynamics

In general, a robotic manipulator system as well as many controlled mechanical systems can be described by the following equation of motion using a minimum set of generalized coordinates $\bar{\mathbf{q}}$:

$$\mathbf{M}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{H}(\bar{\mathbf{q}})\mathbf{u}, \tag{2.5}$$

where $\mathbf{M}(\mathbf{\bar{q}}) \in \mathbb{R}^{n \times n}$ is the positive definite generalized mass matrix of the *n* DoF system. Vector $\mathbf{C}(\mathbf{\bar{q}}, \mathbf{\bar{q}}) \in \mathbb{R}^n$ contains the inertial (centrifugal and Coriolis) terms and all external/active forces, including gravity, spring and damping forces if present in the system.

The control input vector is $\mathbf{u} \in \mathbb{R}^l$ and $\mathbf{H}(\bar{\mathbf{q}}) \in \mathbb{R}^{n \times l}$ is the generalized control input matrix. If the number l of the control inputs is less than the n DoF of the system, then it is called *underactuated*, while if l = n than the system is *fully actuated*. Overactuated systems with l > n are out of scope.

Many studies like [3, 7, 21–25] assume that it is possible to decompose the generalized coordinates $\bar{\mathbf{q}}$ into active (actuated) $\bar{\mathbf{q}}_{a} \in \mathbb{R}^{l}$ and passive (non-actuated) $\bar{\mathbf{q}}_{p} \in \mathbb{R}^{n-l}$ coordinates, leading to:

$$\begin{bmatrix} \mathbf{M}_{aa}(\bar{\mathbf{q}}) & \mathbf{M}_{ap}(\bar{\mathbf{q}}) \\ \mathbf{M}_{pa}(\bar{\mathbf{q}}) & \mathbf{M}_{pp}(\bar{\mathbf{q}}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{a} \\ \ddot{\mathbf{q}}_{p} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{a}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \\ \mathbf{C}_{p}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}.$$
(2.6)

The definition of *Strong Inertial Coupling* is given for symmetric mass matrices in [25], and refers to the coupling between the active and passive coordinates, for which submatrix $\mathbf{M}_{\mathrm{ap}}^{\mathrm{T}}(\bar{\mathbf{q}}) =$ $\mathbf{M}_{\mathrm{pa}}(\bar{\mathbf{q}}) \in \mathbb{R}^{(n-l) \times l}$ is responsible. System (2.6) is said to be Strongly Inertially Coupled if and only if

$$\operatorname{rank}\left(\mathbf{M}_{\mathrm{pa}}(\bar{\mathbf{q}})\right) = n - l \tag{2.7}$$

for any $\bar{\mathbf{q}}$ in the workspace. Note, that (2.7) stands only if $l \ge n-l$, that is, the number of the active coordinates is larger than the number of the passive ones.

The decomposition (2.6) is possible directly only in such specific cases, when the assumption

$$\mathbf{H}(\bar{\mathbf{q}}) = [\mathbf{I} \ \mathbf{0}]^{\mathrm{T}} \tag{2.8}$$

stands in (2.5), where $\mathbf{I} \in \mathbb{R}^{l \times l}$ and $\mathbf{0} \in \mathbb{R}^{(n-l) \times l}$ [3]. The physical meaning of (2.8) is that there are non-zero elements for the active coordinates only. E.g. active and passive decomposition is possible in case of the so-called gymnastic robots (group of serial robots with actuated and passive joints) introduced in [25].

However, the full state feedback linearization cannot be carried out for (2.6), because of the lack of control in the second, passive part of the equations. This part forms a so-called second-order non-holonomic constraint for the system. Based on the literature, a partial feedback linearization for serial robots is summarized in Section 2.4.

In general cases when the assumption (2.8) is not satisfied, we can transform the system (2.5) into a form similar to (2.6) via the projection of the system into the null-space of the input matrix $\mathbf{H}(\bar{\mathbf{q}})$. Let us consider the null-space projection matrix $\mathbf{V}(\bar{\mathbf{q}}) \in \mathbb{R}^{(n-l) \times n}$ of $\mathbf{H}(\bar{\mathbf{q}})$ as:

$$\mathbf{V}(\bar{\mathbf{q}}) = \operatorname{null} \left(\mathbf{H}^{\mathrm{T}}(\bar{\mathbf{q}}) \right)^{\mathrm{T}}.$$
(2.9)

With (2.9), the n-l dimensional passive part of the equation of motion (2.5), also named as *internal dynamics*, can be reformulated as:

$$\mathbf{V}(\bar{\mathbf{q}})\mathbf{M}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}} + \mathbf{V}(\bar{\mathbf{q}})\mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{0}.$$
(2.10)

If we apply the idea of the Moore – Penrose pseudo-inverse then the l dimensional active part of the equation of motion (2.5) can also be derived in the form:

$$\mathbf{H}^{\dagger}(\bar{\mathbf{q}}) \left[\mathbf{M}(\bar{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) - \mathbf{Q}(\bar{\mathbf{q}}) \right] = \mathbf{u}, \qquad (2.11)$$

where the Moore – Penrose pseudo-inverse $\mathbf{H}^{\dagger}(\bar{\mathbf{q}}) \in \mathbb{R}^{l \times n}$ of the input matrix can be calculated as:

$$\mathbf{H}^{\dagger}(\bar{\mathbf{q}}) = \left(\mathbf{H}^{\mathrm{T}}(\bar{\mathbf{q}})\mathbf{W}^{-1}\mathbf{H}(\bar{\mathbf{q}})\right)^{-1}\mathbf{H}(\bar{\mathbf{q}})^{\mathrm{T}}\mathbf{W}^{-1}, \qquad (2.12)$$

with the application of the weight matrix \mathbf{W} . Since this weight matrix can be chosen optionally, the pseudo inverse is not unique in general [26, 27]. Its simplest and most commonly used form is

$$\mathbf{H}^{\dagger}(\bar{\mathbf{q}}) = \left(\mathbf{H}^{\mathrm{T}}(\bar{\mathbf{q}})\mathbf{H}(\bar{\mathbf{q}})\right)^{-1}\mathbf{H}(\bar{\mathbf{q}})^{\mathrm{T}}, \qquad (2.13)$$

where the weight matrix is chosen to be identity.

The splitting into active and passive parts of the equation of motion is even more difficult in the presence of geometric constraint equations, when mechanically complex robotic structures are modeled by non-minimum set of descriptor coordinates. A possible solution is to project the equation of motion into the subspace of kinematically possible motions [28]. After this projection, we end up with an equation of motion of the form of (2.5) and any control technique developed for underactuated systems (e.g., partial feedback linearization [3, 19, 22], computed desired computed torque control [7, 21]) can be applied. However the sequence of projections may require large computational effort. Consequently, the other possibility is to apply the control algorithm directly for a constrained system, which will be detailed in the subsequent sections.

2.3 Relative degree for feedback linearization of SISO systems

Based on the literature [19, 29, 30], the feedback linearization and the idea of the relative degree will be overviewed here in the simple case of single-input single-output (SISO) systems written in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \qquad (2.14)$$

$$y = h(\mathbf{x}),\tag{2.15}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}$ is a scalar output and $u \in \mathbb{R}$ a scalar input. The goal of the feedback linearization is to find a coordinate transformation $\mathbf{z} = \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^n$ with which the control law can be given in the following form:

$$u = a(\mathbf{x}) + b(\mathbf{x})v, \tag{2.16}$$

where v is now the scalar synthetic input. The goal is to realize a linear input-output mapping between the new synthetic input v and the output y by (2.16). Then an outer-loop control strategy for the resulting linear control system can be applied.

To ensure that the transformed system is an equivalent representation of the original system, the transformation $\mathbf{z} = \mathbf{\Phi}(\mathbf{x})$ must be invertible (bijective), and both the transformation and its inverse $\mathbf{x} = \mathbf{\Phi}^{-1}(\mathbf{z})$ must be smooth so that differentiability in the original coordinate system is preserved in the new coordinate system.

The goal of feedback linearization is to produce a transformed system of which the states are the output y and its first n-1 derivatives. For the sake of efficient notation, the literature introduces the Lie derivatives, which is shortly overviewed here. Let us consider the time derivative of the output:

$$\dot{y} = \frac{d}{dt}h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}}.$$
(2.17)

Considering (2.14), we obtain:

$$\dot{y} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) + \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) u \,.$$
(2.18)

The Lie derivatives of $h(\mathbf{x})$ along $f(\mathbf{x})$ and along $g(\mathbf{x})$ are defined by

$$L_f h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}), \qquad (2.19)$$

$$L_g h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}), \qquad (2.20)$$

respectively, where $\partial/\partial \mathbf{x}$ denotes the gradient with respect to \mathbf{x} . Using the notation of these Lie derivatives, the time derivative of the output is

$$\dot{y} = L_f h(\mathbf{x}) + L_g h(\mathbf{x}) u \,. \tag{2.21}$$

The relative degree is an integer that shows how the input u influences the feedback linearized system, which is composed by a state vector consisting of the output y and its first n-1 derivatives. A SISO system given by (2.14) and (2.15) is said to have relative degree r at a point \mathbf{x}_0 if [19, 29],

$$L_g L_f^k h(\mathbf{x}) = 0, \quad \forall k < r - 1$$
(2.22)

$$L_g L_f^{r-1} h(\mathbf{x}_0) \neq 0.$$
 (2.23)

Considering the definition of relative degree in light of the expression of the time derivative of the output y, we can consider the relative degree of our system (2.14) and (2.15) to be the number of times we have to differentiate the output y before the input u appears explicitly. Note that for multi-input multi-output (MIMO) systems, the relative degree can be defined pairwise for all the possible input-output pairs.



FIGURE 2.1. The position control of a linear 1 DoF system

As an example, the relative degree will be calculated for simple mechanical systems in Section 3.4. Additionally, the simplest possible mechanical example is illustrated here in Fig. 2.1, where the control force F provides the position control of the mass m_0 . The position is controlled, thus we can write that the desired system output is y = x. According to Newton's Law, the dynamics of the system is represented by $m_0\ddot{x} = F$. Introducing the state vector $\mathbf{x} = [x_1 \ x_2]$, the resulting SISO system can be formulated in the form of (2.14) and (2.15) by the following equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_0 \end{bmatrix} F, \qquad (2.24)$$

$$y = x_1, \qquad (2.25)$$

with $\mathbf{f}(\mathbf{x}) = [x_2 \ 0]^{\mathrm{T}}$, $\mathbf{g}(\mathbf{x}) = [0 \ 1/m_0]^{\mathrm{T}}$, $h(\mathbf{x}) = x_1$ and u = F. We follow the definition of relative degree given by (2.22) and (2.23). For k = 0 we obtain:

$$L_g h(\mathbf{x}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1/m_0 \end{bmatrix} = 0.$$
(2.26)

For k = 1 first we have to calculate $L_f h(\mathbf{x})$ and then $L_g L_f h(\mathbf{x})$ is determined:

$$L_f h(\mathbf{x}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ 0 \end{bmatrix} = x_2, \qquad (2.27)$$

$$L_g L_f h(\mathbf{x}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1/m_0 \end{bmatrix} = 1/m_0.$$
(2.28)

Since $L_g L_f^k h(\mathbf{x}_0)$ is not zero for k = 1 = r - 1, the relative degree is r = 2.

Here, an alternative way can also be used to calculate r. Considering (2.24) and (2.25), we can observe that the second derivative of the output is in direct relationship with the control input F, while the zeroth and first are not:

$$y = x_1, \qquad (2.29)$$

$$\dot{y} = \dot{x}_1 = x_2 \,, \tag{2.30}$$

$$\ddot{y} = \dot{x}_2 = F/m_0.$$
 (2.31)

In other words, we have to differentiate the output twice before the input appears in it explicitly, that is, the relative degree is r = 2. Relative degree r = 2 is quite general for most of the controlled mechanical systems, however relative degree higher than 2 is also possible depending on the topological structure. More complex cases, where the realtive degree is higher than 2, will be discussed in Section 3.4.

2.4 Partial feedback linearization of collocated and non-collocated serial robots

A special group of manipulators are the serial ones, which is also true in the case of underactuated manipulators. Reference [25] presents the partial feedback linearization of underactuated serial manipulators, of which three types are distinguished as shown in Fig. 2.2. In general case, the robot is described by the minimum set generalized coordinates $\bar{\mathbf{q}} \in \mathbb{R}^n$ and actuated by l < n number of actuators, consequently the robot has n - l number of passive DoF in any sequence of the joints. Each actuator is supposed to actuate a single DoF. According to [25], an underactuated serial robot is *upper actuated* if the first l joint are actuated and in case of *lower actuated* systems the last l joint is actuated one. Then the coordinates are written as $\bar{\mathbf{q}} = [\bar{\mathbf{q}}_a \ \bar{\mathbf{q}}_p]^T$, where coordinate vector $\bar{\mathbf{q}}_a \in \mathbb{R}^l$ and $\bar{\mathbf{q}}_p \in \mathbb{R}^{n-l}$ corresponds to the active and passive joints respectively. The above partitioning allows to write the equations of motion in the form of (2.6) with symmetric mass matrix.



FIGURE 2.2. Serial underactuated robots: general (a), upper actuated (b) and lower actuated (c)

The partial feedback linearization is given only for the so-called *collocated* and *non-collocated* systems, which are special cases. In collocated case, the outputs are the active joint coordinates:

$$\mathbf{y}_{\mathbf{a}} = \bar{\mathbf{q}}_{\mathbf{a}} \in \mathbb{R}^{l} \,, \tag{2.32}$$

while in the non-collocated case the passive joint coordinates are the outputs

$$\mathbf{y}_{\mathrm{p}} = \bar{\mathbf{q}}_{\mathrm{p}} \in \mathbb{R}^{n-l}, \qquad (2.33)$$

For non-collocated systems, the partial feedback linearization is only possible if strong inertial coupling is present between the active and passive coordinates, which condition was already given by (2.7) in Section 2.2.

For both collocated and non-collocated cases the accelerations for the passive coordinates are expressed from (2.6) as the function of $\mathbf{\bar{q}}$, $\mathbf{\bar{q}}$ and $\mathbf{\ddot{q}}_{a}$:

$$\ddot{\mathbf{q}}_{\mathrm{p}} = -\mathbf{M}_{\mathrm{pp}}^{-1}(\bar{\mathbf{q}}) \left(\mathbf{M}_{\mathrm{pa}}(\bar{\mathbf{q}}) \ddot{\mathbf{q}}_{\mathrm{a}} + \mathbf{C}_{\mathrm{p}}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \right) \,. \tag{2.34}$$

Then acceleration $\ddot{\mathbf{q}}_{p}$ is substituted back into the active part of equation (2.6), from which one obtains an *l* dimensional differential equation system:

$$\tilde{\mathbf{M}}_{\mathrm{aa}}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}}_{\mathrm{a}} + \tilde{\mathbf{C}}_{\mathrm{a}}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{u}\,, \tag{2.35}$$

where the symmetric and positive definite matrix $\tilde{\mathbf{M}}_{aa}(\bar{\mathbf{q}})$ and vector $\tilde{\mathbf{C}}_{a}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}})$ are calculated as:

$$\tilde{\mathbf{M}}_{aa}(\bar{\mathbf{q}}) = \mathbf{M}_{aa}(\bar{\mathbf{q}}) - \mathbf{M}_{ap}(\bar{\mathbf{q}})\mathbf{M}_{pp}^{-1}(\bar{\mathbf{q}})\mathbf{M}_{ap}^{\mathrm{T}}(\bar{\mathbf{q}}), \qquad (2.36)$$

$$\tilde{\mathbf{C}}_{a}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{C}_{a}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) - \mathbf{M}_{ap}(\bar{\mathbf{q}})\mathbf{M}_{pp}^{-1}(\bar{\mathbf{q}})\mathbf{C}_{p}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}).$$
(2.37)

At this point, we can define a feedback linearizing controller with synthetic input $\mathbf{v}_a = \ddot{\mathbf{q}}_a$ in the form:

$$\mathbf{u} = \tilde{\mathbf{M}}_{aa}(\bar{\mathbf{q}})\mathbf{v}_{a} + \tilde{\mathbf{C}}_{a}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}).$$
(2.38)

Considering again the passive part of the equation of motion (2.6) and the feedback linearizing controller (2.38), the complete system now is given by the equations:

$$\mathbf{M}_{\rm pp}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}}_{\rm p} + \mathbf{C}_{\rm p}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = -\mathbf{M}_{\rm pa}(\bar{\mathbf{q}})\mathbf{v}_{\rm a}\,, \qquad (2.39)$$

$$\ddot{\mathbf{q}}_{\mathbf{a}} = \mathbf{v}_{\mathbf{a}} \,, \tag{2.40}$$

$$\mathbf{y}_{\mathbf{a}} = \bar{\mathbf{q}}_{\mathbf{a}}, \qquad (2.41)$$

from which we can see that the input-output system for synthetic input \mathbf{v}_{a} and for output \mathbf{y}_{a} is linear and second-order. Equations (2.40) and (2.41) also show that the relative degree is 2 for all input-output pair. Equation (2.39) represents the internal dynamics. The synthetic input may be chosen as

$$\mathbf{v}_{a} = \ddot{\mathbf{q}}_{a}^{d} + \mathbf{K}_{D} \left(\dot{\mathbf{q}}_{a}^{d} - \dot{\mathbf{q}}_{a} \right) + \mathbf{K}_{P} \left(\bar{\mathbf{q}}_{a}^{d} - \bar{\mathbf{q}}_{a} \right) , \qquad (2.42)$$

where $\bar{\mathbf{q}}_{a}^{d}$ represents the desired joint trajectories while \mathbf{K}_{P} and \mathbf{K}_{D} are positive definite gain matrices.

From (2.39-2.41), reference [25] shows that a controller can be designed, which provides asymptotically stable error dynamics for collocated systems.

For non-collocated systems a further transformation has to be done, when the outputs are the passive coordinates, that is $\mathbf{y}_{\rm p} = \bar{\mathbf{q}}_{\rm p}$. The condition (2.7) of strong inertial coupling, which is in connection with $\mathbf{M}_{\rm pa}(\bar{\mathbf{q}})$, has to be satisfied. This means that in (2.39), the control input $\mathbf{v}_{\rm a}$ controls the response of $\bar{\mathbf{q}}_{\rm p}$. In those cases the synthetic control input $\mathbf{v}_{\rm a}$ can be expressed from (2.39) using

the pseudo inverse of $\mathbf{M}_{pa}(\bar{\mathbf{q}})$:

$$\mathbf{v}_{a} = -\mathbf{M}_{pa}^{\dagger}(\bar{\mathbf{q}}) \left(\mathbf{M}_{pp}(\bar{\mathbf{q}}) \mathbf{v}_{p} + \mathbf{C}_{p}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \right) , \qquad (2.43)$$

where $\mathbf{v}_{p} = \ddot{\mathbf{q}}_{p}$ is the new synthetic input. With the above transformation, system (2.39-2.41) has a new representation

$$\ddot{\mathbf{q}}_{\mathbf{p}} = \mathbf{v}_{\mathbf{p}} \,, \tag{2.44}$$

$$\ddot{\bar{\mathbf{q}}}_{a} = -\mathbf{M}_{pa}^{\dagger}(\bar{\mathbf{q}}) \left(\mathbf{M}_{pp}(\bar{\mathbf{q}}) \mathbf{v}_{p} + \mathbf{C}_{p}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \right) , \qquad (2.45)$$

$$\mathbf{y}_{\mathbf{p}} = \bar{\mathbf{q}}_{\mathbf{p}} \,, \tag{2.46}$$

which is applicable for non-collocated systems. Now, the passive coordinates are decomposed from the rest of the system, which is input-output linearized. After that, a locally asymptotically stable controller can be designed.

In the next section a similar method is overviewed from the literature, which does not require that each actuator directly actuates a single DoF.

2.5 Computed desired computed torque control

As already mentioned in Section 2.1, the computed desired computed torque control (CDCTC) method for underactuated systems is introduced in [7] for dynamical systems that are modeled by minimum set of generalized coordinates and consequently, the equations of motion form a system of ordinary differential equations (ODE) defined by (2.5).

The phrase "computed desired" means that the uncontrolled coordinates cannot be arbitrarily prescribed but they can be calculated from the internal dynamics of the controlled system. Contrarily, the controlled coordinates are prescribed. If the synthetic input is assumed in the form of (2.3):

$$\mathbf{v} = \mathbf{M}(\bar{\mathbf{q}}^d)\ddot{\mathbf{q}}^d + \mathbf{K}_{\mathrm{P}}(\bar{\mathbf{q}}^d - \bar{\mathbf{q}}) + \mathbf{K}_{\mathrm{D}}(\dot{\bar{\mathbf{q}}}^d - \dot{\bar{\mathbf{q}}}), \qquad (2.47)$$

where $\mathbf{K}_{\rm P}$ and $\mathbf{K}_{\rm D}$ are the gain matrices of the linear compensator, the control law corresponding to (2.1) has to satisfy

$$\mathbf{H}(\bar{\mathbf{q}}^{d})\mathbf{u} = \mathbf{C}(\bar{\mathbf{q}}^{d}, \dot{\bar{\mathbf{q}}}^{d}) + \mathbf{v}.$$
(2.48)

Thus, the control law that eliminates the error of the controlled coordinates at $t \mapsto \infty$ satisfies the equation:

$$\mathbf{H}(\bar{\mathbf{q}}^{d})\mathbf{u} = \mathbf{M}(\bar{\mathbf{q}}^{d})\ddot{\mathbf{q}}^{d} + \mathbf{C}(\bar{\mathbf{q}}^{d}, \dot{\bar{\mathbf{q}}}^{d}) + \mathbf{K}_{P}(\bar{\mathbf{q}}^{d} - \bar{\mathbf{q}}) + \mathbf{K}_{D}(\dot{\bar{\mathbf{q}}}^{d} - \dot{\bar{\mathbf{q}}}).$$
(2.49)

Equation (2.49) has to be solved for the control input \mathbf{u} and for the uncontrolled subset of the desired generalized coordinates $\bar{\mathbf{q}}^d$. The basic idea is to use the null-space \mathbf{N} of the input matrix \mathbf{H} to project the equations into the space of the uncontrolled motion:

$$\mathbf{0} = \mathbf{N}(\bar{\mathbf{q}}^{d})^{\mathrm{T}} \left(\mathbf{M}(\bar{\mathbf{q}}^{d}) \ddot{\bar{\mathbf{q}}}^{d} + \mathbf{C}(\bar{\mathbf{q}}^{d}, \dot{\bar{\mathbf{q}}}^{d}) + \mathbf{K}_{\mathrm{P}}(\bar{\mathbf{q}}^{d} - \bar{\mathbf{q}}) + \mathbf{K}_{\mathrm{D}}(\dot{\bar{\mathbf{q}}}^{d} - \dot{\bar{\mathbf{q}}}) \right) .$$
(2.50)

This can be solved for the uncontrolled desired coordinates while the generalized coordinates $\bar{\mathbf{q}}$ and velocities $\dot{\bar{\mathbf{q}}}$ appearing in the linear compensator are measured values. The above step of the method shows, that a set of dynamic equations are needed to calculate the uncontrolled part of the generalized coordinates, which means that the inverse kinematics calculation is not possible to accomplish without the consideration of the dynamics of the underactuated system. If we know the uncontrolled desired coordinates, the control inputs can be determined by:

$$\mathbf{u} = \mathbf{H}(\bar{\mathbf{q}}^{d})^{\dagger} \left(\mathbf{M}(\bar{\mathbf{q}}^{d}) \ddot{\bar{\mathbf{q}}}^{d} + \mathbf{C}(\bar{\mathbf{q}}^{d}, \dot{\bar{\mathbf{q}}}^{d}) + \mathbf{K}_{\mathrm{P}}(\bar{\mathbf{q}}^{d} - \bar{\mathbf{q}}) + \mathbf{K}_{\mathrm{D}}(\dot{\bar{\mathbf{q}}}^{d} - \dot{\bar{\mathbf{q}}}) \right),$$
(2.51)

where $\mathbf{H}(\bar{\mathbf{q}}^d)^{\dagger}$ is the generalised (Moore – Penrose pseudo) inverse of the input matrix $\mathbf{H}(\bar{\mathbf{q}}^d)$ calculated as:

$$\mathbf{H}(\bar{\mathbf{q}}^{d})^{\dagger} = (\mathbf{H}(\bar{\mathbf{q}}^{d})^{\mathrm{T}}\mathbf{H}(\bar{\mathbf{q}}^{d}))^{-1}\mathbf{H}(\bar{\mathbf{q}}^{d})^{\mathrm{T}}.$$
(2.52)

It is possible to adopt the CDCTC method for constrained systems described by redundant set of descriptor coordinates where the equations of motion constitute a DAE. The solution requires an additional projection [28] that results an ODE. After this projection the CDCTC method can be applied again. However, this approach is computationally too expensive for online applications due to the repeated projections. Sections 3.2 and 3.3 will introduce and discuss methods to overcome these difficulties.

Chapter 3

Computed torque control of systems with redundant coordinates

As introduced in Chapter 2, the CTC method is an efficient technique for trajectory tracking control of robot manipulators, which needs the inverse dynamics calculation of the controlled dynamical system in each sampling period. A special group of controlled mechanical systems is formed by the underactuated systems, in which the number of independent control inputs is less than the DoF of the system. In these systems, the inverse dynamics calculation is a challenging task, because the inverse calculation leads to the solution [6, 31, 32] of differential-algebraic equations (DAE) [33, 34]. In case of dynamical systems modeled by redundant descriptor coordinates, the equation of motion is originally a system of DAE, and this makes the control intricate [6, 13, 14, 31, 35].

After a summary of the mathematical background, this Chapter introduces some new methods in order to implement CTC for underactuated multibody systems. As it was shown in the literature review in Chaper 2, most of the existing control algorithms are based on an ODE model of the system, either by using minimum set of generalized coordinates, or by eliminating the constraining forces represented by Lagrange multipliers. For the approaches presented in this Chapter, the transformation to an ODE model is not needed and the related advantages will be explained.

3.1 Mathematical background

Several dynamical systems, especially the ones with closed kinematic loops have complex dynamics, which may hardly be modeled using conventional robotic approaches. In order to avoid numerically expensive computations, these complex robotic structures are generally modeled by redundant (or with alternative terminology: non-minimum set, or dependent) descriptor coordinates instead of the minimum set of generalized coordinates used in the Lagrangian approach [36]. A possible modeling technique is based on the natural (Cartesian) coordinates to describe the configuration of the robot. The number of descriptor coordinates is larger than the DoF, thus a set of algebraic equations has to be used to represent the corresponding geometric constraints. In this approach, the mathematical model of the controlled dynamical structure itself is a system of DAE.

3.1.1 Problem formulation

The CTC method is to be generalized for underactuated systems described by non-minimum set of descriptor coordinates. In this case, geometric constraint equations provide the connections between

the redundant descriptor coordinates. Thus, the equation of motion of such systems is given in DAE form. It will be shown that CTC algorithms can directly be applied for these DAE systems.

Using non-minimum set of descriptor coordinates $\mathbf{q} \in \mathbb{R}^n$, the equation of motion subjected to the geometric constraints is written in the following general form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\mathbf{u}, \qquad (3.1)$$

$$\boldsymbol{\varphi}(\mathbf{q},t) = \mathbf{0}\,,\tag{3.2}$$

which is a DAE [8, 36]. Equation (3.1) is the Lagrangian equation of motion of the first kind, where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is a positive definite mass matrix. The descriptor coordinates are chosen intuitively, but if they are chosen properly like in case of the use of the so-called natural coordinates (see [36]), this mass matrix is a constant matrix $\mathbf{M}(\mathbf{q}) \equiv \mathbf{M}$. This will be a relevant observation when the advantages of modeling by DAE are listed. Vector $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ contains the inertial, gyroscopic, Coriolis terms and all external forces, including gravity, spring and damping forces if present. The holonomic and rheonomic geometric constraints are represented by $\varphi(\mathbf{q}, t) \in \mathbb{R}^m$ and $\varphi_{\mathbf{q}}(\mathbf{q}, t) = \partial \varphi(\mathbf{q})/\partial \mathbf{q} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix associated with these geometric constraints. The corresponding Lagrange multipliers are collected in the time dependent vector $\boldsymbol{\lambda} \in \mathbb{R}^m$. Consequently, the system has n - m DoF.

The *l* dimensional control input vector is $\mathbf{u} \in \mathbb{R}^l$ and $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$ is the generalized control input matrix. If the number *l* of the control inputs is less than the n - m DoF of the system, then it is called *underactuated*, while if l = n - m than the system is *fully actuated*.

The task of the manipulator is defined in the form of holonomic and rheonomic constraint equations called *servo-constraints* or *control-constraints* [6, 13, 20, 31, 32, 37–41]. The use of servoconstraints enables to handle them similarly to the geometric constraints (3.2), and gives the possibility to mathematically formulate any kind of manipulator tasks. The servo-constraint equation with the servo-constraint vector $\boldsymbol{\sigma}(\mathbf{q}, t) \in \mathbb{R}^l$ can be written as:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \mathbf{0}.\tag{3.3}$$

We assume that the investigated underactuated system has desired outputs of the same number l as inputs. In spite of the fact that the inverse dynamical calculation leads to the solution of a system of DAE, the desired control inputs can be determined uniquely by the method of computed torques [6, 14, 35]. So the dimension of the servo-constraint vector is also l that means that the number l of control inputs can uniquely be determined for a prescribed task.

Reference [28] mentions that the classical Lagrange multipliers technique works only for independent constraints, where the constraint Jacobian is a full row rank matrix. Considering this, we assume that the servo-constraints are *linearly independent*. Besides, we assume that they are also *consistent*, that is, there are no contradictory constraints, and they can be satisfied with *bounded control input*.

After the introduction of servo-constraint equations, the number n of independent descriptor coordinates are constrained by the same number n = m + l constraint equations in fully actuated cases. When n > m + l in underactuated systems, a part of the dynamics is independent from the geometric and the servo-constraints, which is also called *uncontrolled dynamics*.

The goal is to determine the desired values of the descriptor coordinates in \mathbf{q} , the input vector \mathbf{u} and adjunctively the vector $\boldsymbol{\lambda}$ of Lagrange multipliers, which satisfy the DAE system (3.1), (3.2)
and (3.3). While in some simple cases this goal can be achieved analytically, numerical methods have to be used in practice.

In the following subsections, the application of CTC for fully actuated systems, the different forms of servo-constraints, and the collocated/non-collocated systems are presented and introduced.

3.1.1.1 Servo-constraint based CTC for fully actuated systems

For a general overview of the difference between the inverse dynamic calculation of fully actuated and underactuated systems, let us consider an unconstrained dynamical system with the complementary servo-constraint equation:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{H}(\bar{\mathbf{q}})\mathbf{u}, \tag{3.4}$$

$$\boldsymbol{\sigma}(\bar{\mathbf{q}},t) = \mathbf{0},\tag{3.5}$$

where $\bar{\mathbf{q}} \in \mathbb{R}^n$ is the vector of minimum set generalized coordinates. The desired values in $\bar{\mathbf{q}}$ can be obtained from the servo-constraint equation (3.5) as the function of time. However, the servoconstraint vector $\boldsymbol{\sigma}(\bar{\mathbf{q}}, t)$ is usually a nonlinear function of $\bar{\mathbf{q}}$, so numerical methods should be applied. With this, the control input can easily be calculated, because in case of unconstrained, fully actuated systems the control input matrix $\mathbf{H}(\bar{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is invertible [15]:

$$\mathbf{u} = \mathbf{H}^{-1}(\bar{\mathbf{q}}) \left[\mathbf{M} \ddot{\bar{\mathbf{q}}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \right].$$
(3.6)

The inverse dynamical calculation becomes slightly more challenging if the fully actuated system is described by non-minimum set coordinates, and geometric constraints are introduced, as (3.1) and (3.2) show. In such case the geometric constraint equation (3.2) and the servo-constraint equation (3.3) are both needed to obtain the desired values of the descriptor coordinates $\mathbf{q} \in \mathbb{R}^n$. In contrast with the unconstrained systems, here, the control input can not be calculated with the inverse of $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$, because it is not square matrix. The control algorithms introduced in sections 3.2 and 3.3 will resolve this probem.

3.1.1.2 Alternative servo-constraint formulations

The general form of the servo-constraint is given by (3.3). In the literature some different formalisms can be found for special cases. In [6, 14, 20] the following form is used:

$$\boldsymbol{\sigma}(\mathbf{q}, \mathbf{p}(t)) = \mathbf{0},\tag{3.7}$$

where the dependence of the servo-constraints on the desired system output $\mathbf{p}(t)$ is emphasized, that is $\mathbf{p}(t)$ describes the desired trajectory of a certain point and/or the desired orientation of the end-effector. The servo-constraint may have an even more specific form if the following separation is possible:

$$\boldsymbol{\sigma}(\mathbf{q}, \mathbf{p}(t)) = \mathbf{h}(\mathbf{q}) - \mathbf{p}(t). \tag{3.8}$$

Vector $\mathbf{h}(\mathbf{q})$ provides the prescribed system outputs as the function of the descriptor coordinates. Clearly, this formalism is more specific than (3.3) and even more than (3.7), but it will be used later in Section 6.4.2. In the most specific cases, the separation of the descriptor coordinates is possible into *controlled* and *uncontrolled* coordinates as introduced in reference [7, 21]. In these cases the coordinates whose trajectories are prescribed in time via the task definition are called controlled coordinates. The rest of the descriptor coordinates, called uncontrolled ones, must be calculated with respect to the dynamics of the system, and there is no direct restriction for them from the side of the task description of the manipulator. However, in general, we cannot say that there is a set of coordinates which are prescribed by the task. Still, in some cases, the servo-constraints and a well chosen subset of geometric constraints can be solved for the controlled coordinates is intuitive, the choice is not obvious in case of complex systems. This situation is similar to the case when we choose minimum set of generalized coordinates for a mechanical system in order to obtain the simplest possible system of ODEs, or when we choose the non-minimum set of descriptor coordinates in order to be able to implement CTC as shown in Section 3.4.

Summarizing, if we can find a proper separation of controlled and uncontrolled coordinates, then the task can be defined as

$$\mathbf{q}_{\rm c} = \mathbf{q}_{\rm c}^{\rm d}(t),\tag{3.9}$$

where the superscript d refers to the desired trajectory denoted by $\mathbf{p}(t)$ in the more general formalism (3.8). In the formulation (3.9), the controlled coordinates are prescribed functions of time.

The advantage of all the above special cases (3.7), (3.8) and (3.9) is that large amount of on-line computation time can be saved if the time dependence of the controlled coordinates can be expressed directly.

For the partitioning of the descriptor coordinates, [7] introduces the task dependent selector matrices \mathbf{S}_{c} and \mathbf{S}_{u} with which the controlled and uncontrolled coordinates can be separated:

$$\mathbf{q}_{c} = \mathbf{S}_{c}^{\mathrm{T}} \mathbf{q}, \qquad (3.10)$$

$$\mathbf{q}_{\mathrm{u}} = \mathbf{S}_{\mathrm{u}}^{\mathrm{T}} \mathbf{q}. \tag{3.11}$$

Then the vector of descriptor coordinates is reassembled as

$$\mathbf{q} = \mathbf{S}_{c}\mathbf{q}_{c} + \mathbf{S}_{u}\mathbf{q}_{u}.$$
(3.12)

When CTC is applied, the goal is reduced to the determination of the desired values of the uncontrolled coordinates in \mathbf{q}_{u} , the input vector \mathbf{u} and adjunctively the Lagrange multipliers $\boldsymbol{\lambda}$ that satisfy the DAE system (3.1), (3.2) and (3.9).

This formalism of separated controlled and uncontrolled coordinates is used in the direct discretization based CTC method explained in details in Section 3.2.1.

3.1.1.3 Examples for servo-constraint formulations

Consider the planar crane-like robot depicted in Fig. 3.1 together with its free-body-diagrams. Choose the non-minimum set of descriptor coordinates

$$\mathbf{q} = \begin{bmatrix} x_2 & z_2 & x_3 & z_3 & x_4 & z_4 \end{bmatrix}^{\mathrm{T}}$$
(3.13)

with the geometric constraint

$$\varphi(\mathbf{q}) = \left[(x_3 - x_4)^2 + (z_3 - z_4)^2 - L_{34}^2 \right] \,. \tag{3.14}$$

which leaves 6 - 1 = 5 DoF for the system. This is a simplified structure of the Acroboter platform which will be discussed in details in Section 6. The robot is actuated by means of l = 4 actuators that include three cable winches and one ducted fan leading to the control input vector

$$\mathbf{u} = \begin{bmatrix} F_{12} & F_{23} & F_{24} & F_T \end{bmatrix}^{\mathrm{T}} . \tag{3.15}$$

The payload has 3 DoF, the cable connector has another 2 DoF, but the maximum number of task can only be l = 4, that is the number of the actuators. If the position of the mass centre P_C of the payload is prescribed together with the height h_{cc} of the cable connector P₂ above P_C and the tilt angle α of the payload then the corresponding servo-constraint assumes the form of (3.3):

$$\boldsymbol{\sigma}(\mathbf{q},t) = \begin{bmatrix} \frac{1}{2}(x_3 + x_4) - x^{\mathrm{d}}(t) \\ \frac{1}{2}(z_3 + z_4) - z^{\mathrm{d}}(t) \\ z_2 - \frac{1}{2}(z_3 + z_4) - h_{\mathrm{cc}}^{\mathrm{d}}(t) \\ z_3 - z_4 - L_{34}\sin\alpha^{\mathrm{d}}(t) \end{bmatrix}, \qquad (3.16)$$

where superscript d refers to desired values.



FIGURE 3.1. Simplified structure (a) and free-body-diagrams (b) of the crane-like robot platform Acroboter

The servo-constraint can also be presented in the specific form of (3.8) with the separated functions:

$$\mathbf{h}(\mathbf{q}) = \begin{bmatrix} \frac{1}{2}(x_3 + x_4) & \frac{1}{2}(z_3 + z_4) & z_2 - \frac{1}{2}(z_3 + z_4) & z_3 - z_4 \end{bmatrix}^{\mathrm{T}},$$
(3.17)

$$\mathbf{p}(t) = \begin{bmatrix} x^{\mathrm{d}}(t) & z^{\mathrm{d}}(t) & h_{\mathrm{cc}}^{\mathrm{d}}(t) & L_{34}\sin\alpha^{\mathrm{d}}(t) \end{bmatrix}^{\mathrm{T}},$$
(3.18)

The task can also be defined by means of selected controlled and uncontrolled coordinates as in (3.12) with selector matrices

$$\mathbf{S}_{c} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{S}_{u} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad (3.19)$$

leading to

$$\mathbf{q}_{c} = \begin{bmatrix} z_{2} & x_{3} & z_{3} & z_{4} \end{bmatrix}^{\mathrm{T}}, \qquad (3.20)$$

$$\mathbf{q}_{\mathbf{u}} = \begin{bmatrix} x_2 & x_4 \end{bmatrix}^{\mathsf{T}} . \tag{3.21}$$

In this representation the task can be defined by means of the desired values of the controlled coordinates

$$\mathbf{q}_{c}^{d} = \begin{bmatrix} z^{d}(t) + h_{cc}^{d}(t) & x^{d}(t) - \frac{L_{34}}{2}\cos\alpha^{d}(t) & z^{d}(t) + \frac{L_{34}}{2}\sin\alpha^{d}(t) & z^{d}(t) - \frac{L_{34}}{2}\sin\alpha^{d}(t) \end{bmatrix}^{T}.$$
(3.22)

It would be also possible to select x_4 as a controlled coordinate instead of x_3 . As these examples show, several approaches are feasible for the representation of the tasks in underactuated systems, and the control strategy will be strongly affected by the appropriate choice of the geometrical description of the multibody system.

3.1.1.4 Collocated and non-collocated cases

References [7, 21, 25] use the notion of collocated and non-collocated cases within the control of underactuated systems. These terms are in connection with the separation into controlled coordinates $\mathbf{q}_{\rm c}$ and uncontrolled coordinates $\mathbf{q}_{\rm u}$ (see Section 3.1.1.2). Besides this separation, let us distinguish active (actuated) and passive (non-actuated) coordinates denoted by $\mathbf{q}_{\rm a}$ and $\mathbf{q}_{\rm p}$, respectively (see Section 2.2). Reference [21] defines the collocated case by

$$\mathbf{q}_{c} = \mathbf{q}_{a},\tag{3.23}$$

which means that the controlled coordinates are the actuated ones. In the so-called non-collocated case

$$\mathbf{q}_{c} = \mathbf{q}_{p}, \tag{3.24}$$

which means that the controlled coordinates are the passive ones. An important message of references [21, 25] is that the partial feedback linearization and the CTC cannot be carried out in the non-collocated case if there is no strong inertial coupling between the controlled and the active coordinates. The definition of Strong Inertial Coupling was given in [25] and explained in Section 2.2 by means of (2.7).

The importance of the choice of the descriptor coordinates can be recognized here, because this choice does effect the inertial coupling between the coordinates.

Note that the collocated and non-collocated cases are very special ones, because in general, the actuation of the coordinates depends on the mechanical design of the manipulator while the separation between controlled and uncontrolled coordinates depends on the control task; in general, they are not in connection to each other.

3.1.2 Index reduction of differential algebraic equations

As introduced by means of (3.1) and (3.2) in the previous section, the governing equations of multibody dynamical systems are often given in DAE form. An important indicator of the complexity of a DAE system is the so-called *differentiation index* which refers to the number of time differentiations of the algebraic equations after which the algebraic variables will appear as differential variables [33, 34]. Differentiation index 3 is typical in constrained representation of mechanical systems, since the constraints have to be differentiated three times in order to obtain the time derivative of the Lagrange multipliers, consequently the constraints have to be differentiated only twice in order to eliminate the Lagrange multipliers from the equations. Thus, the index reduction of DAE means the subsequent time derivation of the algebraic equations [32, 36, 38–40, 42].

For the presented CTC methods, the formulation of the geometric constraint equation (3.3) is needed at the level of acceleration by differentiating them twice with respect to time, in order to make the acceleration $\ddot{\mathbf{q}}$ appear explicitly. The first and second time derivatives of the geometric constraint are:

$$\varphi_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \varphi_t(\mathbf{q},t) = \mathbf{0}, \qquad (3.25)$$

$$\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q},t)\ddot{\mathbf{q}}+\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}}+\dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t)=\mathbf{0},$$
(3.26)

where subscript \mathbf{q} refers to the gradient calculated with respect to the descriptor coordinate vector, and subscript t refers to the partial (explicit) time derivative.

It is well known [36, 43, 44] that, when the method of Lagrange multipliers is used, the finite difference based numerical solutions become unstable. The geometric constraint equations can be stabilized by the Baumgarte method [36, 43, 44]. The geometric constraint and its first time derivative appears with a proportional and derivative gain parameters α and β :

$$\varphi_{\mathbf{q}}(\mathbf{q},t)\ddot{\mathbf{q}} + \dot{\varphi}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \dot{\varphi}_{t}(\mathbf{q},\dot{\mathbf{q}},t) + 2\alpha[\varphi_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \varphi_{t}(\mathbf{q},t)] + \beta^{2}\varphi(\mathbf{q},t) = \mathbf{0}.$$
(3.27)

The numerical solutions of equation (3.27) can be stabilized with positive α and β values. Instructions for the optimal selection for these scalar parameters are detailed in [44].

It is very common in multibody dynamics to use the dynamic equation (3.1) and the stabilized, acceleration level geometric constraint equation (3.27) in the following hyper matrix form:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}) \\ \boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{H}(\mathbf{q})\mathbf{u} \\ -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q}, \dot{\mathbf{q}}, t) - 2\alpha[\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q}, t)\dot{\mathbf{q}} + \boldsymbol{\varphi}_{t}(\mathbf{q}, t)] - \beta^{2}\boldsymbol{\varphi}(\mathbf{q}, t) \end{bmatrix}.$$
(3.28)

Using (3.28) a standard ODE solver can be used for the numerical integration of the constrained equation of motion, if the control action **u** is already known.

The servo-constraints (3.3) are treated similarly to the geometric constraints that were expressed in velocity and acceleration level as equations (3.25) and (3.26) show. The servo-constraint equation (3.3) can also be differentiated with respect to time. The velocity and acceleration level servo-constraint equations are:

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \boldsymbol{\sigma}_{t}(\mathbf{q},t) = \mathbf{0}, \qquad (3.29)$$

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\ddot{\mathbf{q}} + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \dot{\boldsymbol{\sigma}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) = \mathbf{0}, \tag{3.30}$$

where $\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t) = \partial \boldsymbol{\sigma}(\mathbf{q},t) / \partial \mathbf{q} \in \mathbb{R}^{l \times n}$ is the Jacobian matrix of the servo-constraint and $\boldsymbol{\sigma}_t(\mathbf{q},t) \in \mathbb{R}^l$ is the partial time derivative of the explicitly time dependent part of the servo-constraint.

In the application of the method of Lagrange multipliers, the servo-constraint equations can also be stabilized by the Baumgarte method [32, 36, 43, 44]:

$$\sigma_{\mathbf{q}}(\mathbf{q},t)\ddot{\mathbf{q}} + \dot{\sigma}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} + \dot{\sigma}_{t}(\mathbf{q},\dot{\mathbf{q}},t) + \mathbf{K}_{\alpha}[\sigma_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \sigma_{t}(\mathbf{q},t)] + \mathbf{K}_{\beta}\sigma(\mathbf{q},t) = \mathbf{0}, \qquad (3.31)$$

where $\mathbf{K}_{\alpha} \in \mathbb{R}^{l \times l}$ and $\mathbf{K}_{\beta} \in \mathbb{R}^{l \times l}$ are positive definite gain matrices. Note that the method can be simplified if these gain matrices are substituted by scalar parameters α and β as it appears in the original work of Baumgarte [43].

3.1.3 ODE formulation using redundant coordinates

Control approaches usually consider dynamical systems given in the form of (2.14) and (2.15) SISO systems or in the form of MIMO system, which have the same form except that the control input and the desired system output are multidimensional. The constraining forces, mathematically the Lagrange multipliers, have to be eliminated from the equation of motion (3.1), in order to apply most of the control theorems presented in the literature, e.g., [45]. For this elimination, one choice is the *method of Lagrange multipliers* [36]. The geometric constraints at acceleration level (3.26) can be substituted into the equation of motion (3.1). After this, the Lagrange multipliers can be expressed in closed form:

$$\boldsymbol{\lambda} = \left(\boldsymbol{\varphi}_{\mathbf{q}} \mathbf{M}^{-1} \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}\right)^{-1} \left[\boldsymbol{\varphi}_{\mathbf{q}} \mathbf{M}^{-1} \left(-\mathbf{C} + \mathbf{H} \mathbf{u}\right) + \dot{\boldsymbol{\varphi}}_{\mathbf{q}} \dot{\mathbf{q}} + \dot{\boldsymbol{\varphi}}_{t}\right].$$
(3.32)

After substituting (3.32) back into the equation of motion (3.1), the acceleration $\ddot{\mathbf{q}}$ can be expressed directly, as equation (3.28) also shows. Note, however, that the numerical solution of the resulting ODE is unstable, so this approach cannot be used for simulations. This is useful however for the calculation of the control input at a definite time instant, but it is computationally time consuming for on-line calculations.

An other possibility to transform the equation of motion into ODE form is the projection of the equation of motion (3.1) into the proper subspaces [28]. Let us consider the decomposition of the variation of the descriptor coordinates $\delta \mathbf{q}$ into the admissible $\delta \mathbf{q}_{a}$ and the constrained $\delta \mathbf{q}_{c}$ parts:

$$\delta \mathbf{q} = \delta \mathbf{q}_{\mathrm{a}} + \delta \mathbf{q}_{\mathrm{c}} \tag{3.33}$$

where $\delta \mathbf{q}_{a}$ is responsible for the motion admitted by the geometric constraints, while $\delta \mathbf{q}_{c}$ satisfies the geometric constraints. With the assumption that the geometric constraints $\varphi(\mathbf{q})$ does not depend on time explicitly, from the time derivative of the constraint equation (3.2), we can write that

$$\boldsymbol{\varphi}_{\mathbf{q}} \delta \dot{\mathbf{q}} = \mathbf{0}. \tag{3.34}$$

The vector $\delta \dot{\mathbf{q}}$ is the difference of two possible velocities admitted by the geometric constraints, so it is the virtual velocity in the classical sense. Since the constraint Jacobian $\varphi_{\mathbf{q}}$ is composed by the gradient vectors of the geometric constraints, it is true that:

$$\boldsymbol{\varphi}_{\mathbf{q}} \delta \dot{\mathbf{q}}_c = \mathbf{0}. \tag{3.35}$$

Considering also (3.33) and (3.34), one obtains:

$$\varphi_{\mathbf{q}}\delta\dot{\mathbf{q}}_{\mathrm{a}} = \mathbf{0}.\tag{3.36}$$

This is satisfied if $\delta \dot{\mathbf{q}}_{\mathrm{a}}$ is in the null-space of $\boldsymbol{\varphi}_{\mathbf{q}}$ defined as:

$$\delta \dot{\mathbf{q}}_{\mathrm{a}} = \mathbf{P}_{\mathrm{a}} \delta \dot{\mathbf{q}},\tag{3.37}$$

$$\mathbf{P}_{\mathbf{a}} = \mathbf{I} - \boldsymbol{\varphi}_{\mathbf{q}}^{\dagger} \boldsymbol{\varphi}_{\mathbf{q}}, \tag{3.38}$$

where $\varphi_{\mathbf{q}}^{\dagger}$ is the Moore – Penrose pseudoinverse of the constraint Jacobian and \mathbf{P}_{a} projects onto the generalized directions that are admissible with the geometric constraints. After this projection, we can construct the equation of the motion in the subspace admitted by the geometric constraints:

$$\mathbf{P}_{\mathbf{a}}^{\mathrm{T}}\left[\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} - \mathbf{H}\mathbf{u}\right] = \mathbf{0}.$$
(3.39)

Note that the calculation of the pseudoinverse $\varphi_{\mathbf{q}}^{\dagger}$ can lead to physically incorrect results depending on the units of the descriptor coordinates in \mathbf{q} . In [28] a modified pseudoinverse calculation was introduced to calculate the projection matrix $\mathbf{P}_{\mathbf{a}}$:

$$\tilde{\boldsymbol{\varphi}}_{\mathbf{q}}^{\dagger} = \mathbf{L}^{-1} (\boldsymbol{\varphi}_{\mathbf{q}} \mathbf{L}^{-1})^{\dagger}, \qquad (3.40)$$

$$\mathbf{P}_{\mathrm{a}} = \mathbf{I} - \tilde{\boldsymbol{\varphi}}_{\mathbf{q}}^{\dagger} \boldsymbol{\varphi}_{\mathbf{q}}, \tag{3.41}$$

where \mathbf{L} is the Cholesky decomposition of the mass matrix \mathbf{M} .

Both methods summarized briefly in this subsection make it possible to use CTC methods for systems defined with geometric constraints, but their application in on-line control algorithms are computationally highly demanding, resulting in low sampling frequencies. The subsequent sections present alternative approaches leading to realistic applications in possible on-line control algorithms.

3.2 Extension of backward Euler discretization

Instead of the intricate application of the existing methods (like the partial feedback linearization in Section 2.4, or the CDCTC in Section 2.5, both combined with the ODE formulation using redundant coordinates in Section 3.1.3), an alternative possibility is to solve the inverse dynamic problem via the direct backward Euler discretization of the DAE system (3.1), (3.2) and (3.3) [14, 32, 46, 47].

The backward Euler method requires the solution of a system of nonlinear algebraic equations in each time step, which can be solved by Newton–Raphson iteration efficiently. A general formalism is proposed for the analytical calculation of the Jacobian of the Newton–Raphson method. The results are also presented in the form of a case study for the Acroboter system and confirmed by numerical simulations. The numerical results are compared with analytical calculations in specific cases.

3.2.1 Separation of controlled coordinates

Section 3.1.1.2 explains the way of separation of the descriptor coordinates \mathbf{q} into controlled and uncontrolled subsets [14] by means of the equations (3.10) and (3.11).

The basic idea is that the backward Euler discretization is applied for the DAE system (3.1) and (3.2), and the resulting set of equations is solved by Newton–Raphson method for the desired actuator forces, the uncontrolled coordinates and the constraining forces. Again, we assume that the task is defined such that the control input **u** will be bounded.

As it was already written in Section 2.1, the control law is formulated by using a synthetic input $\mathbf{v} = \ddot{\mathbf{q}}$:

$$\mathbf{M}(\mathbf{q}^{\mathrm{d}})\mathbf{v} + \mathbf{C}(\mathbf{q}^{\mathrm{d}}, \dot{\mathbf{q}}^{\mathrm{d}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}^{\mathrm{d}}, t)\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q}^{\mathrm{d}})\mathbf{u}, \qquad (3.42)$$

$$\boldsymbol{\varphi}(\mathbf{q}^{\mathrm{d}},t) = \mathbf{0}.\tag{3.43}$$

In order to achieve the convergence of the error between the desired and measured coordinate values to zero, the synthetic input is chosen to:

$$\mathbf{v} = \ddot{\mathbf{q}}^d + \mathbf{K}_D(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) - \mathbf{K}_P(\mathbf{q}^d - \mathbf{q})$$
(3.44)

as in (2.3). Here \mathbf{q} is the measured and \mathbf{q}^{d} is the desired descriptor coordinate vector. The desired descriptor coordinates are known from the controlled part \mathbf{q}_{c}^{d} while the uncontrolled part \mathbf{q}_{u}^{d} will depend on the dynamics and so the coordinates of \mathbf{q}_{u}^{d} are unknown. The measured values of the coordinates of \mathbf{q} appear only in the linear compensator. With the appropriate choice of the positive definit matrices \mathbf{K}_{P} and \mathbf{K}_{D} the error signal converges to zero.

Introduce the velocity vectors $\mathbf{y}^{d} = \dot{\mathbf{q}}^{d}$, $\mathbf{y}_{c}^{d} = \dot{\mathbf{q}}_{c}^{d}$, $\mathbf{y}_{u}^{d} = \dot{\mathbf{q}}_{u}^{d}$ for the full velocity vector, the controlled and the uncontrolled velocity vectors, respectively. Then the first-order form of equations (3.42), (3.43) and (3.44) can be derived after the decomposition of the controlled and uncontrolled coordinates:

$$\dot{\mathbf{q}}_c^{\mathrm{d}} = \mathbf{y}_c^{\mathrm{d}},\tag{3.45}$$

$$\dot{\mathbf{q}}_{\mathbf{u}}^{\mathrm{d}} = \mathbf{y}_{\mathbf{u}}^{\mathrm{d}}, \tag{3.46}$$

$$\dot{\mathbf{y}}_{c}^{d} = \mathbf{S}_{c}^{\mathrm{T}} \mathbf{M}(\mathbf{q}^{\mathrm{d}})^{-1} \left(-\mathbf{C}(\mathbf{q}^{\mathrm{d}}, \mathbf{y}^{\mathrm{d}}) - \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}^{\mathrm{d}}, t) \boldsymbol{\lambda} + \mathbf{H}(\mathbf{q}^{\mathrm{d}}) \mathbf{u} \right) - \mathbf{S}_{c}^{\mathrm{T}} \left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}^{\mathrm{d}} - \mathbf{q}) + \mathbf{K}_{\mathrm{D}}(\mathbf{y}^{\mathrm{d}} - \dot{\mathbf{q}}) \right),$$
(3.47)

$$\dot{\mathbf{y}}_{u}^{d} = \mathbf{S}_{u}^{\mathrm{T}} \mathbf{M}(\mathbf{q}^{d})^{-1} \left(-\mathbf{C}(\mathbf{q}^{d}, \mathbf{y}^{d}) - \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}^{d}, t) \boldsymbol{\lambda} + \mathbf{H}(\mathbf{q}^{d}) \mathbf{u} \right) - \mathbf{S}_{u}^{\mathrm{T}} \left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}^{d} - \mathbf{q}) + \mathbf{K}_{\mathrm{D}}(\mathbf{y}^{d} - \dot{\mathbf{q}}) \right),$$
(3.48)

$$\mathbf{0} = \boldsymbol{\varphi}(\mathbf{q}^{\mathrm{d}}, t), \tag{3.49}$$

which provides the control law as a solution for \mathbf{u} . In the meantime, the constraining forces $\boldsymbol{\lambda}$ and the uncontrolled desired coordinate vector \mathbf{q}_{u}^{d} and velocity vector \mathbf{y}_{u}^{d} are also obtained as a solution of these equations.

All these solutions including the control input are determined by means of the direct backward Euler discretization, which is applied for the general differential equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$ in the form $\mathbf{x}_i - \mathbf{x}_{i-1} = h\mathbf{f}(\mathbf{x}_i)$ where $\mathbf{x}_i = \mathbf{x}(t_i)$, i = 1, 2, ... and h is an appropriately chosen time step. Since the desired values of the controlled coordinates are given by the prescribed task, equation (3.45) is identically true. Then equations (3.46)-(3.49) are discretized by the above described scheme of the backward Euler method:

$$\mathbf{q}_{u,i}^{d} - \mathbf{q}_{u,i-1}^{d} = h \mathbf{y}_{u,i}^{d}, \qquad (3.50)$$
$$\mathbf{y}_{u,i}^{d} - \mathbf{y}_{u,i-1}^{d} = h \mathbf{S}_{u}^{\mathrm{T}} \mathbf{M}(\mathbf{q}_{i}^{d})^{-1} \left(-\mathbf{C}(\mathbf{q}_{i}^{d}, \mathbf{y}_{i}^{d}) - \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{d}, t_{i}) \boldsymbol{\lambda}_{i} + \mathbf{H}(\mathbf{q}_{i}^{d}) \mathbf{u}_{i} \right) - h \mathbf{S}_{u}^{\mathrm{T}} \left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}_{i}^{d} - \mathbf{q}_{i}) + \mathbf{K}_{\mathrm{D}}(\mathbf{y}_{i}^{d} - \dot{\mathbf{q}}_{i}) \right), \qquad (3.51)$$

$$\mathbf{0} \;=\; - \dot{\mathbf{y}}_{c,i}^{\mathrm{d}} + \mathbf{S}_{\mathrm{c}}^{\mathrm{T}} \mathbf{M}(\mathbf{q}_{i}^{\mathrm{d}})^{-1} \left(- \mathbf{C}(\mathbf{q}_{i}^{\mathrm{d}},\mathbf{y}_{i}^{\mathrm{d}}) - oldsymbol{arphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{\mathrm{d}},t_{i}) oldsymbol{\lambda}_{i} + \mathbf{H}(\mathbf{q}_{i}^{\mathrm{d}}) \mathbf{u}_{i}
ight) - \mathbf{v}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{\mathrm{d}},t_{i}) oldsymbol{\lambda}_{i} + \mathbf{H}(\mathbf{q}_{i}^{\mathrm{d}}) \mathbf{u}_{i}
ight) -$$

$$\mathbf{S}_{\mathrm{c}}^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}_{i}^{\mathrm{d}}-\mathbf{q}_{i})+\mathbf{K}_{\mathrm{D}}(\mathbf{y}_{i}^{\mathrm{d}}-\dot{\mathbf{q}}_{i})\right),\tag{3.52}$$

$$\mathbf{0} = \boldsymbol{\varphi}(\mathbf{q}_i^{\mathrm{d}}, t_i) \,. \tag{3.53}$$

Equations (3.50)-(3.53) constitute a system of 2n - l + m number of nonlinear equations for the *i*th value of the desired uncontrolled coordinates $\mathbf{q}_{u,i}^{d} \in \mathbb{R}^{n-l}$, their time derivatives $\mathbf{y}_{u,i}^{d} \in \mathbb{R}^{n-l}$, the control inputs $\mathbf{u}_{i} \in \mathbb{R}^{l}$ and the Lagrange multipliers $\lambda_{i} \in \mathbb{R}^{m}$. Note that $\dot{\mathbf{y}}_{c,i}^{d}$ is a known value from the prescribed task for the controlled coordinates, and the measured values \mathbf{q}_{i} and $\dot{\mathbf{q}}_{i}$ are also known together with the previous values $\mathbf{q}_{u,i-1}^{d}$ and $\mathbf{y}_{u,i-1}^{d}$.

The system of algebraic equations (3.50)-(3.53) can be formulated as a nonlinear function \mathbf{F} of the vector of unknowns \mathbf{z}_i as follows:

$$\mathbf{z}_{i} = \begin{bmatrix} \mathbf{q}_{u,i}^{\mathrm{d}} & \mathbf{y}_{u,i}^{\mathrm{d}} & \mathbf{u}_{i} & \boldsymbol{\lambda}_{i} \end{bmatrix}^{\mathrm{T}},$$

$$\mathbf{F}(\mathbf{z}_{i}) =$$

$$(3.54)$$

$$(3.55)$$

$$\begin{bmatrix} \mathbf{q}_{u,i}^{d} - \mathbf{q}_{u,i-1}^{d} - h\mathbf{y}_{u,i}^{d} \\ \mathbf{y}_{u,i}^{d} - \mathbf{y}_{u,i-1}^{d} + h\mathbf{S}_{u}^{\mathrm{T}}\mathbf{M}^{-1}(\mathbf{q}_{i}^{d})\left(\mathbf{C}(\mathbf{q}_{i}^{d},\mathbf{y}_{i}^{d}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{d},t_{i})\boldsymbol{\lambda}_{i} - \mathbf{H}(\mathbf{q}_{i}^{d})\mathbf{u}_{i}\right) + h\mathbf{S}_{u}^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}_{i}^{d} - \mathbf{q}_{i}) + \mathbf{K}_{\mathrm{D}}(\mathbf{y}_{i}^{d} - \dot{\mathbf{q}}_{i})\right) \\ \dot{\mathbf{y}}_{c,i}^{d} + \mathbf{S}_{c}^{\mathrm{T}}\mathbf{M}^{-1}(\mathbf{q}_{i}^{d})\left(\mathbf{C}(\mathbf{q}_{i}^{d},\mathbf{y}_{i}^{d}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{d},t_{i})\boldsymbol{\lambda}_{i} - \mathbf{H}(\mathbf{q}_{i}^{d})\mathbf{u}_{i}\right) + \mathbf{S}_{c}^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{P}}(\mathbf{q}_{i}^{d} - \mathbf{q}_{i}) + \mathbf{K}_{\mathrm{D}}(\mathbf{y}_{i}^{d} - \dot{\mathbf{q}}_{i})\right) \\ \boldsymbol{\varphi}(\mathbf{q}_{i}^{d},t_{i}) \end{bmatrix}$$

The system of nonlinear equations $\mathbf{F}(\mathbf{z}_i) = \mathbf{0}$ is solved by Newton-Raphson method. The $(j+1)^{\text{st}}$ estimation for the unknown vector \mathbf{z}_i in the *i*th time step is expressed as:

$$\mathbf{z}_{i}^{j+1} = \mathbf{z}_{i}^{j} - \mathbf{J}^{-1}(\mathbf{z}_{i}^{j})\mathbf{F}(\mathbf{z}_{i}^{j}); \quad j = 0, 1, ..., n_{\mathrm{NR}}.$$
(3.56)

If the initial estimations $\mathbf{z}_i^0 = \mathbf{z}_{i-1}^{n_{\text{NR}}}$ are used from the previous time step, then accurate enough results are obtained in a few steps of iterations, that is n_{NR} is usually not larger than 2 or 3.

The iteration can be accelerated even further if the Jacobian of $\mathbf{F}(\mathbf{z}_i)$ is calculated from its analytical formula:

$$\mathbf{J}(\mathbf{z}_{i}) =$$

$$\begin{bmatrix}
\mathbf{I} & -h\mathbf{I} & \mathbf{0} & \mathbf{0} \\
h\mathbf{S}_{u}^{T}\mathbf{M}^{-1}\frac{\partial[\mathbf{C}+\boldsymbol{\varphi}_{q}^{T}\boldsymbol{\lambda}_{i}-\mathbf{H}\mathbf{u}_{i}]}{\partial \mathbf{q}_{u}^{d}} + h\mathbf{S}_{u}^{T}\mathbf{K}_{P}\mathbf{S}_{u} & \mathbf{I}+h\mathbf{S}_{u}^{T}\mathbf{K}_{D}\mathbf{S}_{u} & -h\mathbf{S}_{u}^{T}\mathbf{M}^{-1}\mathbf{H} & h\mathbf{S}_{u}^{T}\mathbf{M}^{-1}\boldsymbol{\varphi}_{q}^{T} \\
\mathbf{S}_{c}^{T}\mathbf{M}^{-1}\frac{\partial[\mathbf{C}+\boldsymbol{\varphi}_{q}^{T}\boldsymbol{\lambda}_{i}-\mathbf{H}\mathbf{u}_{i}]}{\partial \mathbf{q}_{u}^{d}} + \mathbf{S}_{c}^{T}\mathbf{K}_{P}\mathbf{S}_{u} & \mathbf{S}_{c}^{T}\mathbf{K}_{D}\mathbf{S}_{u} & -\mathbf{S}_{c}^{T}\mathbf{M}^{-1}\mathbf{H} & \mathbf{S}_{c}^{T}\mathbf{M}^{-1}\boldsymbol{\varphi}_{q}^{T} \\
\mathbf{\varphi}\mathbf{S}_{u} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix}$$
(3.57)

If the above iteration process is simplified by the use of the initial approximation of the inverse Jacobian in the form:

$$\mathbf{z}_{i}^{j+1} = \mathbf{z}_{i}^{j} - \mathbf{J}^{-1}(\mathbf{z}_{i}^{0})\mathbf{F}(\mathbf{z}_{i}^{j}); \quad j = 0, 1, ..., n_{\mathrm{NR}},$$
(3.58)

we still obtain an accurate enough result in 2-3 steps, and the whole numerical calculation becomes fast enough that it is managable even in on-line control algorithms by providing the control input \mathbf{u}_i within $\mathbf{z}_i^{n_{\text{NR}}}$ in each sampling time instant.

In some cases the Jacobian matrix may be ill-conditioned, but the problem can be handled by singular value decomposition where we leave out the relatively small elements.

3.2.2 Application of servo contraints

If the separation of the coordinates into controlled and uncontrolled parts is not possible as considered in Section 3.2.1, we can still apply the same methodology: the backward Euler discretization of the DAE system (3.1), (3.2) and (3.3) and the Newton-Raphson iteration for the resulting set of nonlinear algebraic equation system.

First, we transform the unconstrained dynamic equation (3.1) into a first-order system via introducing the new variable $\mathbf{y} = \dot{\mathbf{q}}$. Then we consider the geometric constraint equation (3.2) and the stabilized second time derivative (3.31) of the servo-constraint equation (3.3) (see Section 3.1.2). After that, the control law can be obtained for $\mathbf{u} \in \mathbb{R}^l$ as a solution of the 2n + l + m dimensional DAE system:

$$\dot{\mathbf{q}} = \mathbf{y},\tag{3.59}$$

$$\dot{\mathbf{y}} = -\mathbf{M}(\mathbf{q})^{-1} \left[\mathbf{C}(\mathbf{q}, \mathbf{y}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}) \boldsymbol{\lambda} - \mathbf{H}(\mathbf{q}) \mathbf{u} \right], \qquad (3.60)$$

$$\varphi(\mathbf{q}) = \mathbf{0},\tag{3.61}$$

$$oldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{y}}+\dot{oldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q},\mathbf{y},t)\mathbf{y}+\dot{oldsymbol{\sigma}}_{t}(\mathbf{q},\mathbf{y},t)+$$

$$\mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\mathbf{y} + \boldsymbol{\sigma}_{t}(\mathbf{q},t)] + \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q},t) = \mathbf{0}, \qquad (3.62)$$

where $\mathbf{q} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^n$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ are also unknowns. Applying the backward Euler discretization to (3.59-3.62), a system of nonlinear algebraic equations arises in the form:

$$\mathbf{q}_i^{\mathrm{d}} - \mathbf{q}_{i-1} - h\mathbf{y}_i^{\mathrm{d}} = 0, \tag{3.63}$$

$$\mathbf{y}_{i}^{\mathrm{d}} - \mathbf{y}_{i-1} + h\mathbf{M}(\mathbf{q}_{i}^{\mathrm{d}})^{-1} \left[\mathbf{C}(\mathbf{q}_{i}^{\mathrm{d}}, \mathbf{y}_{i}^{\mathrm{d}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}_{i}^{\mathrm{d}})\boldsymbol{\lambda}_{i} - \mathbf{H}(\mathbf{q}_{i}^{\mathrm{d}})\mathbf{u}_{i} \right] = 0, \qquad (3.64)$$

$$\boldsymbol{\varphi}(\mathbf{q}_{i}^{\mathrm{d}}) = \mathbf{0},\tag{3.65}$$

$$\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_{i}^{\mathrm{d}}, t_{i})(\mathbf{y}_{i}^{\mathrm{d}} - \mathbf{y}_{i-1}) + \dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q}_{i}^{\mathrm{d}}, \mathbf{y}_{i}^{\mathrm{d}}, t_{i})\mathbf{y}_{i}^{\mathrm{d}} + \dot{\boldsymbol{\sigma}}_{t}(\mathbf{q}_{i}^{\mathrm{d}}, \mathbf{y}_{i}^{\mathrm{d}}, t_{i}) + \mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q}_{i}^{\mathrm{d}}, t_{i})\mathbf{y}_{i}^{\mathrm{d}} + \boldsymbol{\sigma}_{t}(\mathbf{q}_{i}^{\mathrm{d}}, t_{i})] + \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q}_{i}^{\mathrm{d}}, t_{i}) = \mathbf{0}$$
(3.66)

for the *i*-th value of the desired coordinates \mathbf{q}_i^d , their time derivatives \mathbf{y}_i^d , the control inputs \mathbf{u}_i and the Lagrange multipliers λ_i , while \mathbf{q}_{i-1} and \mathbf{y}_{i-1} are the measured values at the previous time step. It can be formulated as an equation $\mathbf{F}(\mathbf{z}_i) = \mathbf{0}$ of the vector of unknowns:

$$\mathbf{z}_i = [\mathbf{q}_i^{\mathrm{d}} \mathbf{y}_i^{\mathrm{d}} \mathbf{u}_i \boldsymbol{\lambda}_i]^{\mathrm{T}}.$$
(3.67)

The system (3.63-3.66) can be solved by the Newton-Raphson iteration in the same way as explained in the previous subsection (see (3.58)). The only important part of the solution is the control input \mathbf{u}_i since $\mathbf{q}_i^{\rm d}$, $\mathbf{y}_i^{\rm d}$ and λ_i are dropped in the next timestep: the coordinates and velocities are substituted by the measured ones and the constraining forces are not needed at all.

While this method is more general than the case of the separation of controlled and uncontrolled variables, it is computationally more demanding, including the calculation of the Jacobian and its inverse. Still, on-line application can be achieved with this algorithm, too.

3.2.3 Experimental test

The control approach explained in Section 3.2.2 was tested experimentally [48, 49]. The aim of the experiment was to investigate the stability, the accuracy and the computational demands of the presented control approach.

The experimental setup was a fan driven pendulum, also called Aeropendulum [50]. The source of the underactuation is a flexible joint in the Aeropendulum, which can be seen in Fig. 3.2. The mechanical model and the generalized coordinates are presented in Fig. 3.2b; angle θ_1 describes the position of the Aeropendulum, while angle θ_2 gives the relative position of the inverted pendulum attached to the Aeropendulum flexibly. Consequently, the number of DoF is two, and the only control input is the voltage signal given to the fan actuator, so the system is underactuated. The system is collocated in the sense given in Section 3.1.1.4, which means that the angle θ_1 of the actuated DoF is prescribed. The dynamics of the other pendulum is a perturbation the effect of which can be taken into account by the introduced control algorithm in order to reduce the trajectory tracking error.

3.2.3.1 Mechanical model and parameter identification

The mechanical parameters of the experimental rig was unknown, so as a first step we identified them by several preliminary measurements. The parameter identification was carried out by considering the equation of motion of the upper and lower pendulum separately, as it is shown in Fig. 3.3a and



FIGURE 3.2. The experimetal tool (a), the mechanical model (b)

3.3b. The qoverning equations for the lower pendulum are:

$$J_1\ddot{\theta}_1 + B\dot{\theta}_1 + C\text{sgn}(\dot{\theta}_1) + m_1gs_1\sin\theta_1 = M,$$
(3.68)

$$M + DM = Dm_1 g s_1 \sin(\theta_1^0(u)), \tag{3.69}$$

where J_1 is the mass moment of inertia around normal axis through point A of the lower pendulum, B is the parameter responsible for the viscous damping, C is the dry friction torque, $M = l_1 F_T$ is the moment of the fan thrust force about point A, m_1 is the mass of the lower pendulum of length l_1 , s_1 is the distance of the mass centre from point A and $\theta_1^0(u)$ is the static equilibrium angle as a function of a constant u input voltage. The control signal u is given in a 256 bits resolution discrete scale at the digital/analogue converter. Function $\theta_1^0(u)$ was identified at different u values point by point and the data were stored in a lookup table.

Equation (3.69) describes the dynamics of the fan actuator itself. Regarding (3.69), the thrust force $F_{\rm T}$ reaches its nominal value exponentially when a constant input u is applied. Coefficient Dis responsible for the dynamics of the actuator and this is inherited by the actuator moment M. Consequently, the parameter D was identified by means of the response signal $\theta_1(t)$ for unit step excitation in u, while the dry friction torque C was determined by means of the minimal value of the slowly increasing input u where the pendulum started moving.

For parameter identification, the equation of motion of the inverted pendulum was considered in the following form:

$$J_2\ddot{\theta}_2 + k_t\theta_2 - m_2g\frac{l_2}{2}\sin\theta_2 = 0, \qquad (3.70)$$

where J_2 is the mass moment of inertia about the normal axis through point A, k_t is the torsional stiffness of the flexible connection of the upper pendulum of length l_2 and mass m_2 . The upper pendulum was modeled as a homogeneous prismatic bar with mass centre at $s_2 = l_2/2$. The stiffness parameter was identified after measuring the time period of small free oscillations around $\theta_2 = 0$. After the parameter identification, the equations of motion of the flexibly connected pendula are given as:

$$J_1\ddot{\theta}_1 + B\dot{\theta}_1 + C\mathrm{sgn}(\dot{\theta}_1) + m_1gs_1\sin\theta_1 - k_t(\theta_2 - \theta_1) = M, \qquad (3.71)$$

$$\dot{M} + DM = Dm_1 g s_1 \sin(\theta_1^0(u)),$$
 (3.72)

$$J_2\ddot{\theta}_2 + k_t(\theta_2 - \theta_1) - m_2g\frac{l_2}{2}\sin\theta_2 = 0.$$
(3.73)

3.2.3.2 Mechanical model with redundant coordinates

Since the tested control algorithm is applicable for systems with non-minimum set of redundant descriptor coordinates, we derived the mechanical model of the experimental rig with natural coordinates and used this representation in the controller.

As Fig. 3.3c shows, the Cartesian coordinates of the endpoints of the pendula were used as descriptor coordinates: $\mathbf{q} = [x_0, y_0, x_1, y_1, x_2, y_2]^T$. The equations of motion (3.71), (3.72) and (3.73) were reformulated using the algorithm of reference [36] and the resulting governing DAE appeared in the form of (3.1), (3.2) and (3.3). The geometric constraint vector $\boldsymbol{\varphi}(\mathbf{q})$ represents the constant length of the pendulums and the fixed joint at point A,

$$\boldsymbol{\varphi}(\mathbf{q}) = \begin{bmatrix} (x_1 - x_0)^2 + (y_1 - y_0)^2 - l_1^2 \\ (x_2 - x_0)^2 + (y_2 - y_0)^2 - l_2^2 \\ y_0 \\ x_0 \end{bmatrix}, \qquad (3.74)$$

while task is given by the servo-constraint $\sigma(\mathbf{q}, t)$ that prescribes the angle of the lower pendulum by a time dependent function $\theta_1^{\mathrm{d}}(t)$:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \left[\tan \theta_1^{\mathrm{d}}(t) - \frac{x_1 - x_0}{y_0 - y_1} \right], \qquad (3.75)$$

where $\theta_1^d(t)$ is defined by an arc tangent function of time as Fig. 3.4 shows. The n = 6 descriptor coordinates are subjected to m = 4 geometric constraints (3.74), so the system still has n - m = 2DoF out of which one DoF is prescribed by the l = 1 dimensional servo-constraint (3.75).



FIGURE 3.3. Mechanical models applied for the parameter identification (a,b) and the mechanical model using natural coordinates

3.2.3.3 Simulation and experimental results

The control method detailed in Section 3.2.2 and the corresponding discretized equations (3.63-3.66) were applied for the equations of motion of the experimental rig. The angle of the upper pendulum could not be measured because only the lower, actuated pendulum was equipped with an encoder. Because of this, the desired (calculated) values \mathbf{q}_{i-1}^{d} and $\mathbf{y}_{i-1}^{d} = \dot{\mathbf{q}}_{i-1}^{d}$ were used instead of the measured values \mathbf{q}_{i-1} and $\mathbf{y}_{i-1} = \dot{\mathbf{q}}_{i-1}$ in (3.63-3.66) for the not measured coordinates x_0, y_0, x_2 and y_2 .

Fig. 3.4 shows that the realized motion stays quite close to the desired trajectory (upper panel) despite of the low (256 bit) resolution of the control signal (lower panel) and the low sampling frequency of 15 Hz. Note that the long sampling time $h \approx 67 \,\mathrm{ms}$ was due to the poor hardware, and the otherwise complicated control algorithm could also run with a much smaller sampling time. While the computational demand of the control algorithm was perfectly satisfactory here, it can exponentially increase with the complexity, that is with the DoF of the controlled system as it will be discussed in Chapter 6.



FIGURE 3.4. Experimental results

3.3 Extension of Lagrange multipliers method for CTC of underactuated systems

The method of Lagrange multipliers (MLM) is well known regarding the numerical integration of the governing differential-algebraic equation (DAE) of multibody systems [36]. In this section, the MLM is generalized for the servo-constraint based control of underactuated multibody systems described by non-minimum set of descriptor coordinates and corresponding additional geometric constraint equations [32, 38–40, 42].

The basic idea of the method is to join the unconstrained dynamical equation (3.1) to the geometric constraint (3.26) and to the servo-constraint in acceleration level given in (3.31), which is stabilized by the Baumgarte method. We introduce two versions of this basic idea, first the two-steps MLM, and secondly the direct MLM. The two methods can be applied in cases of different assumptions, and they differ in their numerical computation schemes, which is an important issue in online applications. Demonstrative examples will be presented in Section 3.4, and real case application will be described in Chapter 6.

3.3.1 Two-steps approach

In the first part of the two-steps MLM, the null-space projection matrix $\mathbf{V}(\mathbf{q})$ of the input matrix $\mathbf{H}(\mathbf{q})$ is calculated as in (2.9) with which the control input \mathbf{u} can be eliminated from the unconstrained dynamic equation (3.1). This way, we obtain the internal dynamics of the system. This internal dynamics extended with the geometric constraints (3.26) in acceleration level and with the Baumgarte-stabilized servo-constraints (3.31) is expressed in the following hyper matrix form:

$$\begin{bmatrix} \mathbf{V}(\mathbf{q})\mathbf{M}(\mathbf{q}) & \mathbf{V}(\mathbf{q})\boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) \\ \boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} \\ \boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}^{\mathrm{d}} \\ \boldsymbol{\lambda} \end{bmatrix} = \\ \begin{bmatrix} -\mathbf{V}(\mathbf{q})\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \\ -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) \\ -\dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\sigma}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) - \mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \boldsymbol{\sigma}_{t}(\mathbf{q},t)] - \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q},t) \end{bmatrix}.$$
(3.76)

These equations can be solved for the desired descriptor accelerations, and for the Lagrange multipliers, where \mathbf{q} and $\dot{\mathbf{q}}$ come from the measured values. As a second step of the method, the pseudo-inverse of $\mathbf{H}(\mathbf{q})$ is calculated with which the control input can be given as

$$\mathbf{u} = \mathbf{H}^{\dagger}(\mathbf{q}) \left[\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}}^{\mathrm{d}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t) \boldsymbol{\lambda} \right], \qquad (3.77)$$

where $\mathbf{H}^{\dagger}(\mathbf{q})$ is the Moore – Penrose pseudo-inverse of the control input matrix calculated as equation (2.13) shows.

This method works only if the coefficient matrix of equation (3.76) is not singular or it is not close to singular during the numerical calculations, which can be ensured by the appropriate choice of the descriptor coordinates and the phisical realization of the actuation. It will be shown in Section 3.4 that the method may work even in case of non-collocated mechanical systems in the sense of Section 3.1.1.4.

3.3.2 Direct approach

In the one-step MLM, the unconstrained dynamic equation (3.1), the acceleration level geometric constraint equation (3.26) and the acceleration level Baumgarte-stabilized servo-constraint equation (3.31) can be incorporated in a hyper-matrix form as follows:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) & -\mathbf{H}(\mathbf{q}) \\ \boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}^{\mathrm{d}} \\ \boldsymbol{\lambda} \\ \mathbf{u} \end{bmatrix} = \\ \begin{bmatrix} & -\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \\ & -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) \\ & -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) \end{bmatrix}, \quad (3.78)$$

from which the control input \mathbf{u} , the desired acceleration $\ddot{\mathbf{q}}^{d}$ and the vector of Lagrange multipliers $\boldsymbol{\lambda}$ can be calculated as the function of the measured state \mathbf{q} and $\dot{\mathbf{q}}$ of the system. In online applications, only the values of the control input \mathbf{u} are used in each sampling period, while $\ddot{\mathbf{q}}^{d}$ and $\boldsymbol{\lambda}$ are dropped.

This method works only if the coefficient matrix of the unknowns $\ddot{\mathbf{q}}^{d}$, λ and \mathbf{u} in equation (3.78) is not singular or it is not close to singular during the numerical calculations. Note that in case of the so-called non-collocated systems (see Section 3.1.1.4) one must be aware of singularity problems related to this coefficient hyper-matrix.

Clearly, while the direct method can be applied in more general cases, it works numerically slower due to the larger size of the coefficient matrix to be inverted than the two-steps method does. Note also that while the size of the matrices are increased by the use of descriptor coordinates, the elements of these matrices will contain large number of constant values in the generalized mass matrices, which makes the inversion fast even for these large matrices. These considerations justify the development of the otherwise intricate online control algorithms described above.

3.4 Demonstrative comparison of CTC methods

The use of the CTC based algorithms described in Sections 3.2 and 3.3 is studied next. The control algorithms are compared in the case of the simplest possible demonstrative system with special attention to the choice of the descriptor coordinate set [32].

An important step of the control design of robotic manipulators is the choice of the generalized and/or descriptor coordinates. The dynamical modelling of mechanically complex robotic structures via the use of minimum number of generalized coordinates can be inefficient. Several techniques are available in the literature, e.g., the use of reference point coordinates, natural coordinates or mixed coordinates to improve the efficiency of numerical calculations [36]. The numerical computation may be more effective with the use of non-minimum set descriptor coordinates, while the dynamical investigation is more difficult because of the presence of algebraic equations.

In this section, we investigate the CTC of underactuated systems described by non-minimum set of coordinates via a case study which is as simple as possible but catches the main character of such systems.

3.4.1 Mechanical model for case study

The chosen non-trivial dynamical system has the properties described in Section 3.1.1. Consequently, we consider an underactuated n - m = 2 DoF linear system with l = 1 control input only and with one passive DoF. We also introduce m = 1 geometric constraint equation, so the number of descriptor coordinates has to be n = 3. The model shown in Fig. 3.5 is actuated by l = 1 control force F, and the single geometric constraint is represented by a rigid rod of length l_{23} .

The equation of motion formulated in (3.1) and (3.2) can be simplified to be linear by using the stiffness $\mathbf{K} \in \mathbb{R}^{n \times n}$ and the damping $\mathbf{D} \in \mathbb{R}^{n \times n}$ matrices and the vector $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^n$ of gravitational forces, instead of using the general term $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$:

$$\mathbf{M}\ddot{\mathbf{q}}(\mathbf{q}) + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}, \qquad (3.79)$$

$$\varphi(\mathbf{q}) = \mathbf{0}.\tag{3.80}$$



FIGURE 3.5. Underactuated and constrained mechanical model for CTC case study

One has several options to choose descriptor coordinates. The most trivial way is to choose the absolute Cartesian coordinates of the blocks of masses m_1 , m_2 and m_3 as it can be seen in Fig. 3.5:

$$\mathbf{q} = \left[\begin{array}{cc} z_1 & z_2 & z_3 \end{array} \right]^{\mathrm{T}}. \tag{3.81}$$

Then the mass, the stiffness and the damping matrices assume the form

$$\mathbf{M} = \operatorname{diag}\left(m_1, m_2, m_3\right), \qquad (3.82)$$

$$\mathbf{D} = \begin{bmatrix} d & -d & 0 \\ -d & d & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
(3.83)

$$\mathbf{K} = \begin{bmatrix} k & -k & 0 \\ -k & k & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
(3.84)

respectively. The geometric constraint and the constraint Jacobian are:

$$\varphi(\mathbf{q}) = [z_2 - z_3 - l_{23}], \qquad (3.85)$$

$$\boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q}) = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}, \qquad (3.86)$$

respectively. The generalized force vector of the gravitational forces is

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} -m_1g & -m_2g & -m_3g \end{bmatrix}^{\mathrm{T}}.$$
(3.87)

The control input matrix and the control input itself are represented by

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{3.88}$$

$$\mathbf{u} = [F], \tag{3.89}$$

respectively.

As already emphasized in Sections 3.2 and 3.3, the applicability of the control approaches highly depend on the chosen descriptor coordinate set. To represent this, we also choose relative coordinates, which are also typical in robotics, since these are measured by the encoders. Let us introduce the descriptor coordinate vector $\hat{\mathbf{q}} = [z_1 \ \hat{z}_2 \ \hat{z}_3]^{\mathrm{T}}$, where the relative coordinates can be calculated from the previously introduced absolute ones as $\hat{z}_2 = z_2 - z_1$ and $\hat{z}_3 = z_3 - z_2$. The new form of the system matrices are the following:

$$\hat{\mathbf{M}} = \begin{bmatrix} m_1 + m_2 + m_3 & m_2 + m_3 & m_3 \\ m_2 + m_3 & m_2 + m_3 & m_3 \\ m_3 & m_3 & m_3 \end{bmatrix},$$
(3.90)

$$\hat{\mathbf{D}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
(3.91)

$$\hat{\mathbf{K}} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 0 \end{vmatrix},$$
(3.92)

$$\hat{\varphi}(\hat{\mathbf{q}}) = \begin{bmatrix} \hat{z}_3 + l_{23} \end{bmatrix}, \tag{3.93}$$

$$\hat{\boldsymbol{\varphi}}_{\mathbf{q}}(\hat{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \tag{3.94}$$

$$\hat{\mathbf{Q}}(\hat{\mathbf{q}}) = \begin{bmatrix} -(m_1 + m_2 + m_3)g & -(m_2 + m_3)g & -m_3g \end{bmatrix}^{\mathbf{1}}.$$
(3.95)

$$\hat{\mathbf{H}}(\hat{\mathbf{q}}) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{3.96}$$

$$\hat{\mathbf{u}} = [F] \,. \tag{3.97}$$

The most important difference between the two descriptions is that the absolute coordinates do not present inertial coupling between the coordinates due to the diagonal mass matrix, while inertial coupling is present in case of relative coordinates. These trivial observations become especially important because reference [21] shows that in non-collocated cases the CTC can be realized only if there is inertial coupling between the controlled and the active coordinates (see Section 3.1.1.4).

3.4.2 Definition of tasks and corresponding relative degrees

In this section, we give a comparison of the introduced control algorithms by means of numerical simulations for the benchmark problem presented in Figure 3.5. The tested algorithms were the direct backward Euler discretization method (see Section 3.2.2), the two-steps and the direct methods of Lagrange multipliers (see MLM in Section 3.3). There were two different mathematical descriptions of the benchmark system: the case when the absolute Cartesian coordinates are used (table 3.1), and the case when the relative coordinates are used, as explained in details in Section 3.4.1.

Additionally, all numerical simulations are carried out for four tasks. In the first task, the position of mass m_1 is prescribed by $z^{d}(t)$. The mathematical formulation of the servo-constraint is the same in the case of absolute and relative coordinates:

$$\boldsymbol{\sigma}_{\rm c}(\mathbf{q},t) = \hat{\boldsymbol{\sigma}}_{\rm c}(\hat{\mathbf{q}},t) = \left[z_1 - z^{\rm d}(t)\right],\tag{3.98}$$

Since the servo-constraint can be solved for the coordinate of m_1 and the actuator force is also applied to m_1 , this is a collocated case (see Section 3.1.1.4 for the details of collocated and non-collocated cases).

Similarly, the non-collocated case can also be formulated when the prescribed trajectory is related to the passive DoF represented by m_2 and m_3 . When the position of m_2 is prescribed, the servo-constraint has the forms

$$\boldsymbol{\sigma}_{\mathrm{nc2}}(\mathbf{q},t) = \left[z_2 - z^{\mathrm{d}}(t)\right],\tag{3.99}$$

$$\hat{\boldsymbol{\sigma}}_{\mathrm{nc2}}(\hat{\mathbf{q}},t) = \left[z_1 + \hat{z}_2 - z^{\mathrm{d}}(t)\right], \qquad (3.100)$$

in absolute and relative coordinate sets respectively. Call this case "non-collocated type 1". Similarly, when the position of mass m_3 is prescribed ("non-collocated type 2") we can write:

$$\boldsymbol{\sigma}_{\mathrm{nc3}}(\mathbf{q},t) = \left[z_3 - z^{\mathrm{d}}(t)\right],\tag{3.101}$$

$$\hat{\boldsymbol{\sigma}}_{\rm nc3}(\hat{\mathbf{q}},t) = \left[z_1 + \hat{z}_2 + \hat{z}_3 - z^{\rm d}(t)\right].$$
(3.102)

Finally, a "general case" is considered, when the servo-constraint is in relation with both the active and passive DoF. The task is to keep the average position of mass m_1 and m_2 on the prescribed trajectory $z^{d}(t)$.

$$\boldsymbol{\sigma}_{\mathrm{g}}(\mathbf{q},t) = \left[z_1 + z_2 - 2z^{\mathrm{d}}(t)\right],\tag{3.103}$$

$$\hat{\boldsymbol{\sigma}}_{\mathrm{g}}(\hat{\mathbf{q}},t) = \left[2z_1 + \hat{z}_2 - 2z^{\mathrm{d}}(t)\right].$$
(3.104)

In the latter case, there is no unique solution for the servo-constraint, which means that there is no unique choice for controlled and uncontrolled coordinates.

In the following, we calculate the relative degree for both models using absolute and relative coordinates considering the different servo-constraints. First, the state space model with absolute coordinates is derived in the form that was specified by equations (2.14) and (2.15) in Section 2.3. We assume that the geometric constraint (3.85) is satisfied and $l_{23} = 0$. These assumptions lead to a 2 DoF mechanical system with the minimum set of generalized coordinates $\bar{\mathbf{q}} = [z_1 \ z_2]$, where $z_2 = z_3$ and a new mass parameter $m_{23} = m_2 + m_3$ is introduced. The resulting equation of motion reads

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_{23} \end{bmatrix} \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \end{bmatrix} + \begin{bmatrix} d & -d \\ -d & d \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} + \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} m_1 g \\ m_{23} g \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, (3.105)$$

which is rearranged in first-order normal form after introducing the new variables $x_1 = z_1$, $x_2 = z_2$, $x_3 = \dot{z}_1$ and $x_4 = \dot{z}_2$:

$$\dot{x}_1 = x_3,$$
 (3.106)

$$\dot{x}_2 = x_4 \,, \tag{3.107}$$

$$\dot{x}_3 = -\frac{k}{m_1}(x_1 - x_2) - \frac{d}{m_1}(x_3 - x_4) - g + \frac{F}{m_1}, \qquad (3.108)$$

$$\dot{x}_4 = \frac{k}{m_{23}}(x_1 - x_2) + \frac{d}{m_{23}}(x_3 - x_4) - g.$$
(3.109)

In collocated case, when the servo-constraint (3.98) is prescribed, the desired output of the system and its first and second time derivatives are:

$$y = x_1 - z^d$$
, (3.110)

$$\dot{y} = \dot{x}_1 - \dot{z}^d = x_3 - \dot{z}^d \,, \tag{3.111}$$

$$\ddot{y} = \dot{x}_3 - \ddot{z}^d = -\frac{k}{m_1}(x_1 - x_2) - \frac{d}{m_1}(x_3 - x_4) - g - \frac{F}{m_1} - \ddot{z}^d, \qquad (3.112)$$

from which we can observe that the control input F appears in the second time derivative of the output y, thus the relative degree is $r_c = 2$ in collocated case.

In the non-collocated case, when (3.99) or (3.101) defines the servo-constraint, the desired output and its time derivatives are

$$y = x_2 - z^d$$
, (3.113)

$$\dot{y} = \dot{x}_2 - \dot{z}^d = x_4 - \dot{z}^d,$$
(3.114)

$$\ddot{y} = \dot{x}_4 - \ddot{z}^d = \frac{k}{m_{23}}(x_1 - x_2) + \frac{d}{m_{23}}(x_3 - x_4) - g - \ddot{z}^d, \qquad (3.115)$$

$$\ddot{y} = \frac{k}{m_{23}}(\dot{x}_1 - \dot{x}_2) + \frac{d}{m_{23}}(\dot{x}_3 - \dot{x}_4) - \ddot{z}^{d} = = \frac{k}{m_{23}}(x_3 - x_4) + \frac{d}{m_{23}}\left(-\frac{m_1 + m_{23}}{m_1 m_{23}}\left(k(x_1 - x_2) + d(x_3 - x_4)\right) - \frac{F}{m_1}\right) - \ddot{z}^{d}.$$
(3.116)

The control input appears in the third derivative only, thus the relative degree is $r_{\rm nc} = 3$ in the non-collocated case. Note that the relative degree is 4 if the damping parameter d is zero.

In the general case, the desired system output is written as:

$$y = x_1 + x_2 - 2z^d, (3.117)$$

where the servo-constraint is defined by (3.103). Since it contains variable x_1 similarly to the collocated case, we can conclude that the relative degree is only $r_g = 2$, again.

Now the relative degree for the sysem with relative coordinates is calculated for all the 3 types of servo-constraints. After the elimination of the geometric constraint equation (3.93) and introducing the minimum set of generalized coordinates $\bar{\mathbf{q}} = [z_1 \ \hat{z}_2]$, where $\hat{z}_2 = \hat{z}_3$, we obtain the equation of

motion in the form:

$$\begin{bmatrix} m_1 + m_{23} & m_{23} \\ m_{23} & m_{23} \end{bmatrix} \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} z_1 \\ \dot{z}_2 \end{bmatrix} + \begin{bmatrix} m_1 + m_{23}g \\ m_{23}g \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} (3.118)$$

After introducing the new variables $x_1 = z_1$, $x_2 = \hat{z}_2$, $x_3 = \dot{z}_1$ and $x_4 = \dot{\hat{z}}_2$, the first-order normal form is obtained as:

$$\dot{x}_1 = x_3,$$
 (3.119)

$$\dot{x}_2 = x_4 ,$$
 (3.120)

$$\dot{x}_3 = -\frac{k}{m_1}x_2 - \frac{d}{m_1}x_4 - g + \frac{F}{m_1}, \qquad (3.121)$$

$$\dot{x}_4 = k \frac{m_1 + m_{23}}{m_1 m_{23}} x_2 + d \frac{m_1 + m_{23}}{m_1 m_{23}} x_4 - \frac{F}{m_1}.$$
(3.122)

In collocated case, when the servo-constraint (3.98) is prescribed again, the desired output of the system and its first and second time derivatives are:

$$y = x_1 - z^d \,, \tag{3.123}$$

$$\dot{y} = \dot{x}_1 - \dot{z}^d = x_3 - \dot{z}^d, \qquad (3.124)$$

$$\ddot{y} = \dot{x}_3 - \ddot{z}^d = -\frac{k}{m_1}x_2 - \frac{d}{m_1}x_4 - g + \frac{F}{m_1} - \ddot{z}^d, \qquad (3.125)$$

from which we can observe that the control input F appears in the second time derivative of the output y, thus the relative degree is $\hat{r}_c = 2$ in collocated case.

Similarly, in the non-collocated case of servo-constraint equation (3.100), we have the time derivatives:

$$y = x_1 + x_2 - z^{\mathrm{d}}, \tag{3.126}$$

$$\dot{y} = \dot{x}_1 + \dot{x}_2 - \dot{z}^d = x_3 + x_4 - \dot{z}^d, \qquad (3.127)$$

$$\ddot{y} = \dot{x}_3 + \dot{x}_4 - \ddot{z}^d = -\frac{1}{m_{23}} \left(kx_2 + dx_4 \right) - g - \ddot{z}^d , \qquad (3.128)$$

$$\ddot{y} = -\frac{1}{m_{23}} \left(k\dot{x}_2 + d\dot{x}_4 \right) - \ddot{z}^{d} = -\frac{1}{m_{23}} \left(kx_4 + d \left(k\frac{m_1 + m_{23}}{m_1 m_{23}} x_2 + d\frac{m_1 + m_{23}}{m_1 m_{23}} x_4 - \frac{F}{m_1} \right) \right) - \ddot{z}^{d}.$$
(3.129)

The input F appears again in the 3rd time derivative of the output, thus the relative degree is $\hat{r}_{nc} = 3$. Note that the relative degree is 4 if the damping parameter d is zero.

The general case is given by considering the servo-constraint equation (3.104). The desired system output is

$$y = 2x_1 + x_2 - 2z^d \,. \tag{3.130}$$

Since it contains variable x_1 similarly to the collocated case, we can conclude that the relative degree is $\hat{r}_g = 2$.

3.4.3 Comparison of control strategies by numerical simulations

The mechanical parameters of the model introduced in Section 3.4.1 were chosen to be $m_1 = 1 \text{ kg}$, $m_2 = m_3 = 0.5 \text{ kg}$, k = 10 N/m and d = 1 Ns/m. Sampling time was set to 20 ms. Gain matrices \mathbf{K}_{α} and \mathbf{K}_{β} was tuned by trial and error method in order to drive the violation of the servo-constraint to zero.

Tables 3.1 and 3.2 show the summary of the results of numerical simulations. Three qualitatively different behaviors of the controlled systems are distinguished. In the cases denoted by "A", the controller drives the system into the desired state without any problem, thus the servo-constraint is driven to zero even in the presence of an initial perturbation.

	relative	backward	two-steps	direct
	degree	Euler	MLM	MLM
Collocated	$r_{\rm c} = 2$	А	А	А
Non-collocated, type 1	$r_{\rm nc} = 3$	А	В	С
Non-collocated, type 2	$r_{\rm nc} = 3$	А	В	С
General	$r_{\rm g} = 2$	А	А	А

TABLE 3.1. Benchmark of the control algorithms for absolute coordinates

	relative	backward	two-steps	direct
	degree	Euler	MLM	MLM
Collocated	$\hat{r}_{\rm c} = 2$	А	А	А
Non-collocated, type 1	$\hat{r}_{\rm nc} = 3$	А	А	С
Non-collocated, type 2	$\hat{r}_{\rm nc} = 3$	А	А	С
General	$\hat{r}_{\rm g} = 2$	А	А	А

TABLE 3.2. Benchmark of the control algorithms for relative coordinates

In the cases denoted by "C", the controller encounters singularity problems, and it does not work at all. This problem appears only in the 1-step MLM method, when the coefficient matrix of the unknowns becomes singular (see equation (3.78)).

In cases denoted by "B", the control algorithm seems to work without singularity problems, but the servo-constraint is not driven to zero, consequently, the control is unsuccessful. This occurs in cases of the two-steps MLM method, in spite of the fact that this method is more robust against singularity problems than the direct MLM, since it utilizes null-space projection and pseudo-inverse calculation instead of a complete inverse calculation as shown in (3.76) and (3.77). Still, in the cases "B", equation (3.76) gives zero acceleration for mass m_1 , therefore the computed control force is constant. Because of the null-space projection, the controller loses the information about the desired motion of the passive DoF. This explains, why the two-steps MLM is not succesful in non-collocated cases with absolute descriptor coordinates.

We can conclude that the backward Euler discretization method is successful in all test cases, even in the so-called non-collocated cases. The two-steps MLM is applicable in some cases when the direct MLM fails. We can also observe that the exchange of the descriptor coordinates from absolute to relative ones influences the operation of the control algorithm in two cases: for both non-collocated cases the two-steps MLM works successfully with relative coordinates. Note that the operation of the control algorithms is the same in type 1 and type 2 non-collocated cases. Typical trajectories, and the time history of the servo-constraints and control inputs are presented in Figures 3.6 and 3.7. Figure 3.6 shows a simulation result of type "A" for a non-collocated case described by relative coordinates and controlled by two-steps MLM. Figure 3.7 shows a simulation result of type "B" for non-collocated case with two-steps MLM and absolute coordinates.



FIGURE 3.6. A simulation result of type "A" (successful control): non-collocated case, relative coordinates, two-steps MLM

3.4.4 Concluding remarks

Different control algorithms were detailed for underactuated systems, which are described by nonminimum set of coordinates. These control algorithms were numerically investigated for constrained, underactuated systems with special attention to the choice of the descriptor coordinates of the controlled mechanical system.

All the introduced control algorithms applied the servo-constraint on acceleration level with Baumgarte stabilization, which is widely used in numerical simulation of multibody systems. In the cases of successful control strategies, the simulations showed that the servo-constraints were driven to zero. The detailed study of several combinations of the choices of system coordinates, tasks and control algorithms showed that the proper choice of the descriptor coordinates may result successful control while the same control algorithm may fail with other choices of coordinates. Our study showed that the categories "collocated" and "non-collocated" are in connection with the choice of descriptor coordinates. In practice, it means that in the case of a given robotic system and a given task, the descriptor coordinates can be chosen properly in such a way that the computed torque control algorithm will be successful, otherwise the algorithm may fail. The numerical simulations also represented the robustness of the backward Euler direct discretization of the control law, which works even for non-collocated systems with any choice of coordinates.



FIGURE 3.7. A simulation result of type "B" (unsuccesful control): non-collocated case, absolute coordinates, two-steps MLM

3.5 New results

Thesis 1.

The method of computed torque control was extended for underactuated dynamical systems, where a non-minimum set of descriptor coordinates is used to avoid singularities and nonlinearities in the generalized mass matrix. It was shown that the computed torque control can be realized in real time by means of the backward Euler discretization if the relative degree of the system is two, which is often the case in multibody systems.

Two procedures were developed and compared to each other:

a) In case the controlled and uncontrolled descriptor coordinates can be separated, the calculation of the control torques was carried out in a reduced size algebraic system.

b) In case the separation of the controlled and uncontrolled coordinates is not feasible, the control torque calculation was carried out by means of the coupling of the servo-constraints and the algebraic system resulted by the implicit Euler discretization.

While method b) is more general and easy to implement, the computational demand of method a) is more favourable. The underactuated Aeropendulum rig was used to show experimentally that the CTC can be realized in real time even with the more general control algorithm b).

Related publications: [14, 32, 35, 45–49]

Thesis 2.

The method of computed torque control was generalized for underactuated dynamical systems where inertial coupling appears between the control input and the desired output, and the desired motion is prescribed by means of servo-constraints in a mathematical structure similar to the geometric constraints. It was shown that the computed torque control can be realized in real time by means of the method of Lagrange multipliers even for some non-collocated mechanical systems.

In the differential-algebraic equation model of the system, two procedures were developed using the Baumgarte stabilization for the servo-constraints only. Both procedures are efficient if they are far enough from singularity configurations of the coordinates and the control actions.

It was demonstrated that the control torque can be expressed in an explicit form by means of the null-space projection and the pseudo-inverse of the input matrix. This procedure is computationally more efficient than the application of the direct inversion of the coefficient matrix in the full linear system, which works for collocated mechanical systems only.

Related publications: [32, 38–40, 45]

Chapter 4

Applications for underactuated systems of varying topology

Robotic structures may change their topology during their operation for several reasons. The cause of the topology variation could be related to the variation of the contact points between the robot and the environment, or to the variation in the actually controlled actuators. Accordingly two basic sets of problems will be analysed in this Chapter.

The first study is about the locomotion of a worm-like system [51]. Legged and worm-like locomotion systems are usually underactuated because the actuators are responsible for varying the shape of the locomotion system, and the force that pushes forward the system arises from the ground friction force. In other words there is no direct actuation which moves the body forward. The topology of the studied worm-like locomotion system varies since alternating parts of the system are in connection with the ground at different time instants.

The second topic of Chapter 4 is about the handling of actuator saturation of robotic manipulators by a combination of a fully actuated and an underactuated CTC algorithm. The change of the topology is due to the loss of some of the accessible actuators caused by their saturations. These actuator saturations are the reason why these systems can be considered as underactuated ones.

4.1 Worm-like locomotions

The success of snakes and worms in our planet is mainly based on the simple construction of their body and their robust locomotion technique. They move their body periodically to generate propulsive force and get forward using the interaction with the surrounding environment. The worm can be modeled by means of a simple multibody system containing lumped masses constrained to each other by ideal rigid rods. The periodic motion of the worm body is achieved via the use of an artificial muscle-like actuator system.

4.1.1 Limbless locomotion

The simplest model for limbless locomotion systems, which can be found in the literature, is shown in Fig. 4.1. Two blocks are connected to each other by a linear actuator, and they perform locomotion on the flat ground. The resultant force that pushes the system forward is due to the dry friction of

which the friction coefficient is different in the forward and the backward direction. The asymmetrical friction force is achieved by small spikes placed on the bottom of the blocks. The system is underactuated, since it has two DoF and only one actuator.



FIGURE 4.1. A simple model for limbless locomotion

In biology, much more sophisticated movements are classified named after the worm, the caterpillar and the snake (see Fig. 4.2).



FIGURE 4.2. Locomotion techniques of limbless animals: worm's peristaltic motion (a), caterpillar's locomotion (b), snake's movement (c)

According to [52], the chosen worm model belongs to the *autonomous locomotion systems* because it has internal drives and during the time of the locomotion (t_0, t_0+T) there are only non-driving external forces. Furthermore, neither the center of mass nor any particle of the body remains fixed or runs a cycle in space during that time interval. The non-driving external force is the contact force between the contact points of the worm and the ground. We use only periodic internal drives, thus the locomotion of the selected worm-model is *undulatory locomotion* according to [52].

4.1.2 Mechanical model of a worm-like locomotion

Fig. 4.3. shows the planar mechanical model of a simple worm-like system [51]. The model consists of lumped masses m at points P_i , i = 1, ..., N, connected to each other through N - 1 ideal rigid rods without inertia. The modeling of the contact with the ground, the bending stiffness and actuation of the worm body are detailed in the subsequent sections.



FIGURE 4.3. Mechanical model

4.1.2.1 Parametrization

It is advantageous to use redundant set of coordinates We choose non-minimum set of descriptor coordinates and introduce geometric constraints since these coordinates are dependent. The Cartesian coordinates of lumped masses can be arranged in the descriptor coordinate vector \mathbf{q} :

$$\mathbf{q} = [x_1..x_N, z_1..z_N]^{\mathrm{T}} .$$
(4.1)

The dynamical model can be written in the form of a differential-algebraic equation (DAE), which is adopted from:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, t) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}), \qquad (4.2)$$

$$\varphi(\mathbf{q}) = \mathbf{0}\,,\tag{4.3}$$

which is very similar to the DAE system (3.1) and (3.2) introduced in the problem formulation in Section 3.1.1. The difference is that $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, t) \in \mathbb{R}^{2N}$ is time dependent because the actuation of the worm is included here, which means, that an open-loop control is applied in these cases; there is no feedback. The geometric constraint $\varphi(\mathbf{q}) \in \mathbb{R}^{(N-1)}$ is scleronomic since there is no explicit time dependency. Vector $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^{2N}$ contains gravitational forces.

4.1.2.2 Geometric constraints

The geometric constraint equation (4.3) represents the constant length L_i of the rigid rods connecting the lumped masses. For the sake of simplicity, the geometric constraints are considered to be scleronomic, this means that the length L_i of each rod is constant. The geometric constraint vector is the following:

$$\boldsymbol{\varphi}(\mathbf{q}) = \begin{bmatrix} (x_2 - x_1)^2 + (z_2 - z_1)^2 - L_1^2 \\ \vdots \\ (x_i - x_{i-1})^2 + (z_i - z_{i-1})^2 - L_{i-1}^2 \\ \vdots \\ (x_N - x_{N-1})^2 + (z_N - z_{N-1})^2 - L_{N-1}^2 \end{bmatrix}.$$
(4.4)

From this, the constraint Jacobian $\varphi_{\mathbf{q}}(\mathbf{q})$ can be algorithmically computed in closed form.

4.1.2.3 Stiffness and damping added to the worm body

For the realistic behavior of the model, some stiffness and damping need to be added to the chain-like model. We define an integrated torsional stiffness and torsional damper element, which produce the torque:

$$\tau_i = k\psi_i + b\dot{\psi}_i, \qquad i = 1, \dots, N,$$
(4.5)

where k is the torsional stiffness and b is the damping parameter, while ψ_i is the relative angle of two neighboring rods as shown in Fig. 4.4 and 4.6. After the calculation of the torque τ_i the problem is its representation in terms of the dependent coordinates **q** defined in (4.1). In planar cases, a torque can be replaced by an equivalent pair of forces, **F** and $-\mathbf{F}$, of equal magnitude and opposite directions [36]. Applying this, the torque τ_i acting on rod i and i-1 can be substituted by forces $|\mathbf{F}_i| = \tau_i/|\mathbf{r}_i|$ and $|\mathbf{F}_{i-1}| = \tau_{i-1}/|\mathbf{r}_{i-1}|$ as Fig. 4.4. shows. The derived generalized forces are included in matrix **C** of the equation of motion (4.2).



FIGURE 4.4. The angle of neighboring rods, and the force from stiffness and damping

To compute the torque, ψ_i has to be defined. It would be easy to calculate from the following formula, as [36] recommends:

$$\mathbf{r}_{i-1} \cdot \mathbf{r}_i = L^2 \cos \psi_i,\tag{4.6}$$

with $|\mathbf{r}_{i-1}| = L$ and $|\mathbf{r}_i| = L$. Angle ψ_i can be calculated by the arccos function. However, this function is interpreted in the interval [-1,1] and, because of the numerical rounding problems, the simulations often result values out of this range and the calculation fails. A possible solution is the use of the atan2 function as follows

$$\psi_i = \operatorname{atan2}(r_{i,\xi}; r_{i,\zeta}), \tag{4.7}$$

where atan2 is the two-argument variation of the arctangent function and avoids the problem of division by a small number. Equation (4.7) gives the four quadrant arctangent of the arguments $r_{i,\xi}$ and $r_{i,\zeta}$. Here, $r_{i,\xi}$ and $r_{i,\zeta}$ are the ξ and ζ direction components of \mathbf{r}_i respectively. To get components ξ and ζ , we need to create the rotation matrix \mathbf{R}_i which rotates \mathbf{r}_{i-1} into the local coordinate system (ξ, ζ) from the global system (x, z). \mathbf{R}_i can be created from the direction and normal vector of \mathbf{r}_{i-1} :

$$\mathbf{R}_{i} = \begin{bmatrix} \mathbf{e}_{i} & \mathbf{n}_{i} \end{bmatrix}; \qquad i = 2 \dots N.$$
(4.8)

To calculate the torque according to equation (4.5), the angular velocity $\dot{\psi}_i$ has to be known as well. This can be done using the correlation of the velocity between points P_i and P_{i+1} :

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \boldsymbol{\omega}_i \times \mathbf{r}_i; \qquad i = 2 \dots N, \tag{4.9}$$

where $\boldsymbol{\omega}_i = [0 \ 0 \ \dot{\varphi}_i]^{\mathrm{T}}$ is the angular velocity vector, and φ_i is the absolute angle of the rod, connecting point P_i to P_{i+1} , measured from the horizontal direction, as it is shown in Fig. 4.4. From equation (4.9) the angular velocity $\dot{\varphi}_i$ of each rod can be express, and from this we obtain the relative angular velocity $\dot{\psi}_i = \dot{\varphi}_i - \dot{\varphi}_{i-1}$.

4.1.2.4 Actuation of the worm model

The worm locomotion is achieved by the periodic motion of the body parts, and this periodic motion is generated by periodic internal drives. We mimic artificial muscles between the neighboring segments, and this causes the motion of the worm. The effect of the artificial muscle is simply achieved by offsetting the relaxed angle of the torsional springs with a pre-defined value ψ_i^0 . This working principle is similar to the real skeletal muscles [53]. The design and the representation of angle ψ_i^0 set by a servo motor can be seen in Fig. 4.6. The offset ψ_i^0 of the relaxed angle modifies the torque arising from the torsional spring and damper:

$$\hat{\tau}_i = k(\psi_i - \psi_i^0) + b\dot{\psi}_i, \tag{4.10}$$

where ψ_i^0 is given by a pre-defined periodic control function of time t and segment index i as:

$$\psi_i^0 = A\sin(\omega t + \vartheta i) + c, \tag{4.11}$$

where A is the amplitude of the offset, ω is the angular frequency, ϑ is the phase shift. We can prestress the springs with arbitrary periodic control function (e.g., sine waves). With proper magnitude and phase of the periodic control function (4.11) the worm takes up wave form and periodically stresses and relaxes the torsional springs, which generates a moving wave along the worm body. In appropriate conditions the moving wave propulses the worm forward as it can be seen on Fig. 4.5.



FIGURE 4.5. Worm model performing locomotion

4.1.2.5 Contact with the ground

As we mentioned in Section 4.1.1, the contact with the environment is essential in case of autonomous locomotion systems. We only consider the contact with planar, horizontal surface with Coulomb friction.

The contact force can be split into two parts; the normal component $\mathbf{F}_{g,i}$ is originated from the elastic connection of each particle and the ground, the tangent component $\mathbf{F}_{f,i}$ is the friction force. This friction force propulses the locomotion system forward. The free body diagram of one lumped mass and the forces acting on the ground are shown in Fig. 4.7, where \mathbf{K}_i and \mathbf{K}_{i+1} are constraint forces transmitted by the rods. The elastic connection is modeled with spring of stiffness k_g and damping element of damping factor b_g in the ground. When the lumped mass contacts the ground, the normal component of the contact force $F_{g,i}$ is computed for every lumped mass from the governing equation of the Kelvin-Voigt element:

$$F_{q,i} = -k_q z_i - b_q \dot{z}_i; \quad i = 1, \dots, N.$$
(4.12)

The connection is interrupted in the simulation every time when the $F_{g,i}$ has sign reversal. The friction force is calculated as $F_{f,i} = \mu F_{g,i} \operatorname{sgn}(\dot{x}_i)$; $\dot{x}_i \neq 0$ and it is checked in every time step if adherence has occurred. If so, an event handling is called and coefficient of static friction μ_0 is used.



FIGURE 4.6. Torsional spring and damper (left), offset of the relaxed angle (right)



FIGURE 4.7. Free body diagram of a lumped mass, and forces acting on the ground

4.1.3 Numerical simulation

We use the previously introduced (3.28) hyper-matrix form of the Lagrangian equation of the first kind to solve the equation of motion (4.2) and (4.3), without the feedback control term $\mathbf{H}(\mathbf{q})\mathbf{u}$. In this case instead of calculating the computed torque forces at each time step, we use the algorithm introduced in Chapter 3 for numerical simulation only. Accordingly, the Baumgarte stabilization is used again for the acceleration level of the geometric constraints.

The proper choice of the parameters α and β is very important. A wrong parameter set can cause that the stabilized system is far away from the original mechanical system behavior, or it still remains unstable. In order to chose a stable parameter set, we follow the recommendation of [44], which is based on the stability of a discrete feedback system. The unstable and the stabilized behaviour are shown in Fig. 4.8.



FIGURE 4.8. Constraint violation

4.1.4 Optimization

The aim of the optimization is to gain a proper parameter set that allows us to build a feasible prototype for experiments. Naturally, the speed of the worm is to be maximized.

To determine an effective locomotion of the worm-model within the engineering optimum, we used two different methods to optimize the parameters of the worm-model. The model has many parameters for example: the mass m, the number of the lumped masses N, the damping b and stiffness k of the torsional elements and the four parameters A, ω, θ, c of the control wave (4.11) of the artificial muscles explained in Section 4.1.2.4. According to structural considerations, the mass and the number of the lumped masses are fixed, and the parameters of the ground are set to be realistic. Hence, we consider 5 parameters only with simply scanning a certain range of these parameters. This method is based on the subdivision of the chosen parameter ranges and the calculation of all the possible combinations of the discrete parameter values. After this procedure, 5 dimensional parameter diagrams can be drawn, and the observation of the global effect of the parameter changes is possible, so the range of parameters can be narrowed. We also tried mathematical optimization functions (fsolve, fminsearch) built in the Matlab Optimization Toolkit. With these functions we have done a constrained optimum searching. All methods resulted the same parameter values presented in Table 4.1. Fig. 4.5 shows the corresponding stroboscopic picture of the resulting motion.

name	notation	value	name	notation	value
mass	m	2 kg	magnitude	A	1.3 rad
number of masses	N	20 -	phase shift	θ	5.25 rad
torsional stiffness	k	500 Nm/rad	angular frequency	ω	2 rad/s
torsional damping	b	50 Nms/rad	constant	С	0 rad

TABLE 4.1. Optimized parameters

4.1.5 Concluding remarks

After inspecting the behavior of limbless animals, a model of an autonomous worm-like locomotion system has been developed. The advantage of the model is that it is quite simple and contains only lumped masses and rigid rods, so the equation of motion can be generated in closed form with the proper algorithms. Because of the algorithmic derivation of equations of motion, the generalization of the model to be spatial is straightforward. The contact of the lumped masses with the ground was modeled considering two phenomenon: the dry friction and the visco-elastic behavior. With this, the impulsive dynamics related to the impact between the particles of the model and the ground is considered and simulated as finite dynamics. For the sake of simplicity, the geometric constraints were considered to be scleronomic, but deduction of the mathematical model can be generalized for explicitly time dependent geometric constraints. An optimal and feasible parameter set was determined for the control with a muscle-like actuator system.

To sum up, the worm-like movement was successfully simulated and optimized with a multybody model that was based on the use of redundant set of descriptor coordinates and acceleration level geometric constraints.

4.2 Underactuation induced by actuator saturation

The saturation of actuators is a relevant nonlinearity, and the consideration of the bounded actuator torques during the design of the computed torque control is a challenging task. Here, the saturation of the actuator torques is handled as a temporary reduction of the number of accessible control inputs. Consequently, a manipulator, which is fully actuated, becomes underactuated when intricate combinations of the actuator saturations occur. The cascading of two CTC algorithms is applied to resolve the problem of the temporary underactuation: the classical CTC method is self-interrupted and exchanged to a generalized CTC, which is extended for the control of underactuated systems. The interpretation of the corresponding formalism is applicable even in those cases of multi-body problems where redundant coordinates provide the computationally efficient mathematical description that involves DAE.

Every driver applied in robotic systems has some limitations which are typically speed, power and torque limits. These limitations can be included into the mathematical models. Controlled dynamical systems with limited actuator torque form a special class of nonlinear systems. The actuator saturation may cause essential problems when a manipulator performs trajectory tracking. The risk of actuator saturation is higher, when high accelerations are prescribed.

Several CTC based control approaches can be found in the literature that take into account the limited actuator torques. In [54] a continuous-time predictive control approach is used to derive the nonlinear constrained control law for tracking control in the presence of actuator saturation. The proposed method is limited for those systems that are input-output feedback-linearizable after a specific dynamic expansion. The control minimizes the tracking errors even with saturated actuators. In reference [55], an adaptive full-state feedback controller as well as an exact-model-knowledge output feedback controller are designed, and a comparative numerical analysis is carried out to demonstrate the benefits of the two proposed controller methods. On the basis of the classical CTC method, a composite nonlinear feedback design method is presented in [56] for robot manipulators with bounded torques. The resulting controller consists of two loops. The inner loop is for the full compensation of the manipulator's nonlinear dynamics while the outer loop is the composite nonlinear feedback controller for stabilization and performance enhancement.

The above mentioned control approaches handle the actuator saturation as a nonlinearity of the system. As an alternative approach, actuator saturation can also be described as the decrement of the number of accessible control inputs, which practically means the variation of the manipulator's topology [57].

Let us consider a fully actuated system, where the number of accessible control inputs equals to the DoF of the system. When some of the actuators saturate, the number of accessible control inputs becomes less than the DoF. In this section, we develop a combination of the classical CTC method (see, for example, [15, 16] and Section 2.1) and a CTC method extended for underactuated systems (see [6, 7, 14, 32, 35, 38] and Section 3) in order to handle the actuator saturation. Additionally, the introduced method is proposed for complex multi-body systems which are modeled by redundant descriptor coordinates.

The methods detailed in Section 3 are directly applicable for underactuated systems when control inputs are supposed to be unbounded. The application for the case of bounded inputs is introduced in the subsequent sections.

4.2.1 Combined fully actuated and underactuated algorithm for handling saturation



FIGURE 4.9. Actuator saturation: nonlinear connection between commanded and real control input

In the proposed concept, the control algorithm first calculates the desired control input vector $\mathbf{u} \in \mathbb{R}^l$ for all actuators, and then check whether the value of each control input u_i , $i = 1 \dots l$ exceeds the maximum or minimum limiting values u_i^+ or u_i^- , respectively (see Fig. 4.9). If some actuators saturate, then the number of the non-saturated actuators is reduced to \hat{l} . The saturated actuators are considered to provide a constant torque, and the algorithm calculates again the desired control input $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{l}}$ with $\hat{l} < l$. The number \hat{l} of the still accessible non-saturated control inputs is always less than the number of DoF, so the system is actually underactuated. For the $l - \hat{l}$ number of saturated actuators, the limiting values u_i^+ or u_i^- are commanded by the controller. This operation is repeated in each sampling time step again and again if new and new actuators saturate, but it stops if there are no further actuators to saturate, or if all the actuators are saturated [57].

Equation (4.13) shows the partitioning of the control input vector into saturated and nonsaturated parts:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} = \mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm} + \mathbf{H}(\mathbf{q})\mathbf{T}\hat{\mathbf{u}}, \qquad (4.13)$$

where \mathbf{u}^{\pm} contains the corresponding maximal and minimal values u_i^{\pm} or u_i^{-} , while $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{l}}$ contains the accessible control inputs, $\mathbf{T} \in \mathbb{R}^{l \times \hat{l}}$ and $\mathbf{R} \in \mathbb{R}^{l \times l}$ are selector matrices defined by

$$\mathbf{u} = \mathbf{R}\mathbf{u}^{\pm} + \mathbf{T}\hat{\mathbf{u}} \,. \tag{4.14}$$

In (4.13) and (4.14), the selector matrix \mathbf{R} collects the saturated control inputs, and \mathbf{T} forms the non-saturated (still accessible) control input vector as:

$$\hat{\mathbf{u}} = \mathbf{T}^{\mathrm{T}} \mathbf{u} \,. \tag{4.15}$$

In (4.13), the term $\mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm}$ is a known, constant external force, and the term $\mathbf{H}(\mathbf{q})\mathbf{T}\hat{\mathbf{u}}$ is responsible for the actuation of the system. Thus, we can introduce a reduced size control input matrix $\hat{\mathbf{H}}(\mathbf{q}) \in \mathbb{R}^{n \times \hat{l}}$ for the saturated system as

$$\ddot{\mathbf{H}}(\mathbf{q}) = \mathbf{H}(\mathbf{q})\mathbf{T} \tag{4.16}$$

and the new vector of inertial forces in the form

$$\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{H}(\mathbf{q})\mathbf{R}\mathbf{u}^{\pm}.$$
(4.17)

This way, equation (4.13) assumes the form

$$\mathbf{M}\ddot{\mathbf{q}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} = \hat{\mathbf{H}}(\mathbf{q})\hat{\mathbf{u}}, \qquad (4.18)$$

which is fully compatible with the form (3.1) used in the problem formulation of underactuated systems described by redundant set of coordinates.

It is still true that the inverse calculation can be unique only if the dimension of the servoconstraint vector (that is, the dimension of the task) is equal to the number \hat{l} of accessible control inputs $\hat{\mathbf{u}}$. Consequently a reduced size servo-constraint vector has to be defined for the case when some of the actuators saturate, which practically means the redesign of the desired task [41]. In the saturated cases, we use this reduced size servo-constraint vector $\hat{\boldsymbol{\sigma}}(\mathbf{q},t) \in \mathbb{R}^{\hat{l}}$ instead of the original $\boldsymbol{\sigma}(\mathbf{q},t) \in \mathbb{R}^{l}$ used in (3.3). The dimension reduction [57] of the servo-constraint vector is a critical problem, because the transformation between $\hat{\boldsymbol{\sigma}}(\mathbf{q},t)$ and $\boldsymbol{\sigma}(\mathbf{q},t)$ is not unique: several optimization approaches can be used, which are explained in the following sections describing first specific, then general cases.

4.2.2 Dimension reduction of servo-constraint in specific cases

In the general equation of motion (3.1) of controlled dynamical systems, term $\mathbf{H}(\mathbf{q})\mathbf{u}$ represents the control force. This control force can be viewed as a control reaction force interpreted by means of a Lagrange multiplier $\lambda_{\mathbf{u}} \in \mathbb{R}^l$ associated with the servo-constraint $\boldsymbol{\sigma}(\mathbf{q},t)$ and its Jacobian $\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t)$, (see [6] and [31]). Using this concept, the equation of motion can be rewritten in the form:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q})\boldsymbol{\lambda} + \boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)\boldsymbol{\lambda}_{\mathbf{u}} = \mathbf{0}, \qquad (4.19)$$

considering that

$$\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t)\boldsymbol{\lambda}_{\mathbf{u}} = -\mathbf{H}(\mathbf{q})\mathbf{u}.$$
(4.20)

The connection of the control input \mathbf{u} and the multiplier $\lambda_{\mathbf{u}}$ is trivial if the servo-contstraint Jacobian and the negative control input matrix are just equal:

$$\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) = -\mathbf{H}(\mathbf{q}), \qquad (4.21)$$

$$\mathbf{u} = \boldsymbol{\lambda}_{\mathbf{u}} \,. \tag{4.22}$$

This is called "specific case" when the directions defined by the control input matrix and the servoconstraint Jacobian are the same. The structure of the matrix $\mathbf{H}(\mathbf{q})$ definitely depends on the mechanical design of the system and the chosen set of generalized coordinates, while $\sigma_{\mathbf{q}}(\mathbf{q},t)$ depends on the task description. In these specific cases, the effect of the control inputs on the system coordinates is the same as the effect of the servo-constraints on the motion: the correspondence of the control inputs and the servo-constraints is trivial. Clearly, the above conditions can be satisfied in specific cases only (see examples later in Section 4.2.4).

In the saturated system, the control input matrix of the still non-saturated inputs is defined by equation (4.16). In the specific case, the connection of the reduced servo-constraint Jacobian $\hat{\sigma}_{\mathbf{q}}(\mathbf{q},t)$ and the reduced size control input matrix $\hat{\mathbf{H}}(\mathbf{q})$ should be defined as equation (4.21) does:

$$\hat{\boldsymbol{\sigma}}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) = -\hat{\mathbf{H}}(\mathbf{q}). \qquad (4.23)$$
Considering equation (4.16) we can write:

$$\hat{\boldsymbol{\sigma}}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) = \boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t)\mathbf{T}.$$
(4.24)

From (4.24), the suggested reduced servo-constraint is simply:

$$\hat{\boldsymbol{\sigma}}(\mathbf{q},t) = \mathbf{T}^{\mathrm{T}}\boldsymbol{\sigma}(\mathbf{q},t). \tag{4.25}$$

Note that this transformation is the same as the transformation between the full control input vector and the vector of the non-saturated control inputs in equation (4.15).

4.2.3 Dimension reduction of servo-constraint in general cases

In general cases, the servo-contstraint Jacobian and the negative control input matrix are not equal:

$$\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) \neq -\mathbf{H}(\mathbf{q}), \tag{4.26}$$

which means that the directions defined by the control input matrix and the servo-constraint Jacobian are not necessarily the same. If minimum set of generalized coordinates are used, the transformation between **u** and $\lambda_{\mathbf{u}}$ can be deduced from (4.21) as

$$\boldsymbol{\lambda}_{\mathbf{u}} = -(\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t))^{-1} \mathbf{H}(\mathbf{q}) \mathbf{u} \,. \tag{4.27}$$

The reduced servo-constraint than can be obtained by means of the same transformation:

$$\hat{\boldsymbol{\sigma}}(\mathbf{q},t) = -\mathbf{T}^{\mathrm{T}}\mathbf{H}(\mathbf{q})^{-1}\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t)\boldsymbol{\sigma}(\mathbf{q},t), \qquad (4.28)$$

which simplifies to (4.25) in the special case of (4.21).

If further actuators saturate or redundant descriptor coordinates are used, the same procedure should be repeated for an already underactuated or constrained system, respectively. Since the matrices are non-symmetric in these cases, the transformation between **u** and $\lambda_{\mathbf{u}}$ can be deduced from (4.21) with the use of the Moore – Penrose generalized inverse (pseudo-inverse). Although the transformation is not unique, we can choose, for example the formula:

$$\boldsymbol{\lambda}_{\mathbf{u}} = -(\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t))^{\dagger} \mathbf{H}(\mathbf{q}) \mathbf{u}.$$
(4.29)

The pseudo-inverse calculation itself is not unique either, although the standard optimized calculation

$$(\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}})^{\dagger} = \left(\boldsymbol{\sigma}_{\mathbf{q}}\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}\right)^{-1}\boldsymbol{\sigma}_{\mathbf{q}}$$
(4.30)

is widely used for convenience.

Similarly to the above cases, equation (4.29) generates the reduced servo-constraint in the form

$$\hat{\boldsymbol{\sigma}}(\mathbf{q},t) = \hat{\Gamma}^{\dagger}(\mathbf{q},t)\boldsymbol{\sigma}(\mathbf{q},t), \qquad (4.31)$$

where $\hat{\Gamma}(\mathbf{q}, t)$ is defined as

$$\hat{\Gamma}(\mathbf{q},t) = -(\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t))^{\dagger} \mathbf{H}(\mathbf{q}) \mathbf{T}.$$
(4.32)

Equations (4.31) and (4.32) lead to the following dimension reduction formula:

$$\hat{\boldsymbol{\sigma}} = \left(-(\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}})^{\dagger} \mathbf{H} \mathbf{T} \right)^{\dagger} \boldsymbol{\sigma} \,. \tag{4.33}$$

In the subsequent section, this formula will be used for the dimension reduction of the servo-constraint in the general case.

4.2.4 Case study for an RR manipulator

Numerical simulations were accomplished for a two-link (SCARA-type) manipulator shown in Fig. 4.10, which consists of two homogeneous prismatic bars with parameters $m_1 = 0.2$ kg, $L_1 = 0.4$ m, $m_2 = 0.2$ kg and $L_2 = 0.4$ m, respectively. The manipulator moves in the horizontal plane, thus the effect of the gravitation is canceled. The geometric description and the derivation of the equations of motion



FIGURE 4.10. Mechanical model of the studied RR manipulator

for the chosen system would be straightforward with the minimum set of generalized coordinates α_1 and α_2 . Still, in order to show the applicability of the generalized CTC algorithm, a redundant set of descriptor coordinates is used. Thus, the Cartesian coordinates $\mathbf{q} = [x_1, y_1, x_2, y_2, x_3, y_3]^T$ of the endpoints of the bars are chosen as descriptor coordinates of number n = 6. Consequently 4 dimensional geometric constraint vector is introduced in the form:

$$\boldsymbol{\varphi}(\mathbf{q}) = \begin{bmatrix} x_1 \\ y_1 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_1^2 \\ (x_3 - x_2)^2 + (y_3 - y_2)^2 - L_2^2 \end{bmatrix}.$$
(4.34)

Both joint P₁ and P₂ are actuated, so the l = 2 dimensional control input vector is $\mathbf{u} = [\tau_1 \ \tau_2]^T$ with torques $\tau_{1,2}$. The actuators saturate when the actuator torques reach the limiting values: $u_1^{\pm} = \pm 0.04 \,\mathrm{Nm}$ and $u_2^{\pm} = \pm 0.04 \,\mathrm{Nm}$. Following the derivation detailed in [36], the control input matrix $\mathbf{H}(\mathbf{q})$ is obtained after the virtual substitution of the control torques by pairs of forces applied at the base points:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} -\frac{y_1 - y_2}{L_1^2} & 0\\ \frac{x_1 - x_2}{L_1^2} & 0\\ \frac{y_1 - y_2}{L_1^2} & -\frac{y_2 - y_3}{L_2^2}\\ -\frac{x_1 - x_2}{L_1^2} & \frac{x_2 - x_3}{L_2^2}\\ 0 & \frac{y_2 - y_3}{L_2^2}\\ 0 & -\frac{x_2 - x_3}{L_2^2} \end{bmatrix}.$$
(4.35)

In the following subsections the manipulator is subjected to two tasks defined by two servoconstraints. In the first specific case, the angles α_1 and α_2 of the bars are prescribed in time by $\alpha_1^d(t)$ and $\alpha_2^d(t)$. The desired initial and end configurations are shown in the left panel of Fig. 4.11. In the second case study, a linear path of the endpoint P₃ of the manipulator is prescribed (see the right panel of Fig. 4.11). The transitions from the initial to the end configurations are defined in time by arc tangent functions in both cases.



FIGURE 4.11. The prescribed initial and end configurations in specific (left) and in general (right) cases

4.2.4.1 Specific case

By intuition, we can define the servo-constraint in the form

$$\boldsymbol{\sigma}(\mathbf{q},t) = \begin{bmatrix} -\tan^{-1}\left(\frac{x_1 - x_2}{y_1 - y_2}\right) - \alpha_1^{\mathrm{d}}(t) \\ -\tan^{-1}\left(\frac{x_2 - x_3}{y_2 - y_3}\right) - \alpha_2^{\mathrm{d}}(t) \end{bmatrix}.$$
(4.36)

A lengthy calculation shows that $\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) = -\mathbf{H}(\mathbf{q})$, that is, (4.21) is satisfied. The above intuition can be explained physically since the servo-constraints are related to the angles of the bars, and the control torques also act directly at the same angles.

The simulation results in Fig 4.12 for this specific case are presented for three cases. In case A, the actuator torques do not saturate, the value of the control torque u_1 reaches even 0.13 Nm as it can be seen in the right panel. With the chosen set of proportional and differential gains $\mathbf{K}_{\alpha} = 60 \mathbf{I} 1/\mathrm{s}$, $\mathbf{K}_{\beta} = 48 \mathbf{I} 1/\mathrm{s}^2$ and sampling time $h = 40 \mathrm{ms}$ in equations (3.63-3.66), the servo-constraint violation

was kept under 0.02 rad as shown in the left panel. The fluctuation in the servo-constraint violation at t = 8 s was caused by the highest acceleration demand at the inflection point of the arc tangent time histories.

Case B shows the effect of the actuator saturation when the above introduced control algorithm is not implemented, that is, the control inputs are simply truncated. Significant increment occurs in the servo-constraint violation when the control input u_1 reaches the critical 0.04 Nm saturation value as it can be seen in the graphs of panel B of Fig. 4.12. Note that the second control input u_2 remains under the saturation value all the time.



FIGURE 4.12. Numerical results for specific cases A, B and C: servo-constraint violations (left column) and control inputs (right column)

In case C, the controller switches to the reduced servo-constraint (4.25) with selector matrix

$$\mathbf{T} = \begin{bmatrix} 0\\1 \end{bmatrix} \tag{4.37}$$

during the saturation of the first actuator as explained in Section 4.2.1:

$$\hat{\boldsymbol{\sigma}}(\mathbf{q},t) = \left[-\tan^{-1} \left(\frac{x_2 - x_3}{y_2 - y_3} \right) - \alpha_2^{\mathrm{d}}(t) \right].$$
(4.38)

When the first actuator saturates, the second control input is recalculated as if the system were underactuated, and this causes the increased value of the input u_2 . As a result, the first servoconstraint violation σ_1 is reduced a bit and σ_2 substantially as compared to case B.

The phisical meaning of the use of equation (4.38) during the saturation of the first actuator is that the control effort then focuses on the second servo-constraint which is in connection with the non-saturated and still accessible control input u_2 . One can observe that the second actuator also saturates for a very short time in case C. This shows that the available performance of the actuators are utilized more efficiently than they are in case B.

4.2.4.2 General case

The task of the manipulator is to move the endpoint P_3 of the manipulator from point P_{init} to P_{end} on a prescribed straight line trajectory as shown in right panel of Fig. 4.11. The task is defined by the servo-constraint vector:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \begin{bmatrix} x_3 - x^{\mathrm{d}}(t), \\ y_3 - y^{\mathrm{d}}(t) \end{bmatrix}, \qquad (4.39)$$

where $x^{d}(t)$ is described by an arc tangent function in time and $y^{d}(t) \equiv -0.6$ m is a constant value. Clearly, for the task defined by (4.39) $\mathbf{H}(\mathbf{q}) \neq -\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q}, t)$, so the connection between the servoconstraints and the control inputs is general. The dimension reduction of the servo-constraint vector is carried out by the method explained in Section 4.2.3.

Three cases are considered again, and the corresponding results are shown in Fig. 4.13. In case A, the actuators do not saturate and the value of the control torque u_1 reaches -0.08 Nm. With the same set of control gains and sampling time as above, the initially set servo-constraint errors 4 mm and 12 mm tend to zero in a short time and after that the servo-constraint violation is kept under 2 mm. Case B shows the effect of the actuator saturation when the reduced servo-constraint is not implemented. Significant increment occurs in the servo-constraint violation when the actuator torque u_1 reaches the critical 0.04 Nm. In case C, the controller uses the reduced servo-constraint (4.31) during the actuator saturation. After a larger transient, the servo-constraint violation σ_1 approaches zero much faster than in case B, while σ_2 also decreases somewhat.



FIGURE 4.13. Numerical results for case A, B and C

4.2.5 Concluding remarks

The combination of computed torque control algorithms developed for constrained and underactuated systems was successfully implemented for handling actuator saturation. The developed frame algorithm manages the continuous variation of the number of saturated and non-saturated actuators via employing two inferior CTC algorithms: one for fully actuated and one for underactuated systems. Numerical simulations in a case study presented the efficiency of the proposed method. The developed method requires the temporary dimension reduction of the original servo-constraint either in the specific case, when the servo-constraint Jacobian and the control input matrix are equal or in the general case when they are not equal. While the case study shows moderate improvement in following the prescribed task during the saturation of one or more of the actuators, further research can optimize the servo-constraint reduction further by means of the large number of free parameters appearing in the pseudo inverse calculations in (4.31).

4.3 New results

Thesis 3.

The actuator saturation is considered as a change in the topology of the controlled dynamical system: the fully actuated robot becomes underactuated as some of the actuators saturates. A high level algorithm is developed that switches between the computed torque control algorithm applied for the fully actuated system and another applied for the underactuated one.

A servo-constraint $\hat{\sigma}$ of reduced dimension is introduced for the motion design of the underactuted system, which is generated from the original servo-constraint σ of the fully-actuated system with the help of the transformation:

$$\hat{\boldsymbol{\sigma}} = \left(- (\boldsymbol{\sigma}_{\mathbf{q}}^{\mathrm{T}})^{\dagger} \mathbf{H} \mathbf{T}
ight)^{\dagger} \boldsymbol{\sigma} \,,$$

where $\sigma_{\mathbf{q}}$ is the Jacobian of the servo-constraint, **H** is the control input matrix, **T** is the selector matrix that identifies the non-saturated control inputs and \dagger refers to the pseudo-inverse of the eventually non-quadratic matrices occuring during the reduction process.

With the help of this procedure, the trajectory tracking error of the saturated system is designed and can be optimized.

Related publications: [51, 57]

Chapter 5

Use of parametric excitation for underactuated systems

Consider cases when certain tasks are prescribed for the passive DoF of an underactuated mulytibody system. By augmenting the actuator forces with some periodic excitation for the active DoF, the tasks could be approached even for the passive DoF. Since this periodic excitation at the actuators usually presents some time-periodic parameters in the equations of motion, this kind of forcing is called parametric excitation in classical mechanics. In this sense, parametric excitation could succesfully be used for the control of certain underactuated systems. Two basic scenarios will be discussed in this chapter.

5.1 Pendulum-like robots

There exist already several realizations of indoor/domestic robotic applications. Besides the floor based mobile robots, different ceiling based, crane-like robots have appeared in recent years. A pendulum-like underactuated service robot platform called Acroboter [12] is shown in Fig 5.1a. The swinging unit is equipped with ducted fan actuators and cable winches, thus its 6 DoF can fully be controlled directly. Still, the number of actuators of the full system including also the suspending cable and the climbing unit has l = 10 actuators while the number of DoF is 12, so the system is underactuated. Due to the underactuation, the horizontal position of the cable connector is complicated to control, thus it can have serious oscillations. Additionally, in some conditions, the Arcoboter is planned to operate like a crane, when the fan actuators of the swinging unit are only applied for the elimination of the effect of external perturbations. In this case, the 4 cable winches and the 3 actuators on the climber unit is considered as independent actuators.

A similar but simplified concept is the Winch-Bot that is presented in [58] with structure shown in Fig 5.1b. A cable winch is the only actuator on the robot, that is l = 1, while the end-effector moving in the vertical plane has n = 2 DoF. Similarly to the Acroboter concept, the end-effector can be moved swiftly in a large workspace as an underactured system. Since the cable length can only be actuated, the swinging motion of the end-effector is induced by parametric excitation as the cable length is varied periodically. The control by means of parametric excitation can also be explained from energy viewpoint as the added periodic excitation at the actuator along the active DoF provides additional kinetic energy to the system, which is then transmitted into the passive DoF in a way that the prescribed task is approached as much as possible.



FIGURE 5.1. Acroboter system (a) path of a pendulum-like robot (b)

5.1.1 Task definition

When computed torque control (CTC) is applied for underactuated systems, restrictions may apply for the task definition of the robot. Two approaches are distinguished [59]:

A) one can specify the trajectory of some coordinates in time, but the number of these prescribed coordinates must be equal to the number of control inputs l;

B) the path for all DoF is specified but the time dependent trajectories are not.

In both cases we can obtain unique solution for the inverse dynamical calculation and the CTC inputs.

Use conept B) for the WinchBot in Fig. 5.1b where the task is given by parametric functions for the minimum set of generalized (actually, polar) coordinates $\bar{\mathbf{q}} = [r \vartheta]^{\mathrm{T}}$ in the form:

$$\begin{bmatrix} r\\ \vartheta \end{bmatrix} = \begin{bmatrix} \sqrt{x(p)^2 + y(p)^2}\\ -\tan^{-1}\frac{x(p)}{y(p)} \end{bmatrix}.$$
(5.1)

The endpoint coordinates of the robot are defined by the parametric functions x(p) and y(p) with parameter p, but the time history p(t) cannot be prescribed. From (5.1), $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ can be expressed as a function of the parameter p and its unknown time derivatives \dot{p} and \ddot{p} . With this, the passive part of the equation of motion provides a differential equation for p(t); its solution is detailed in [58].

The same task can also be defined by the servo-constraint (see (3.3)) in case of concept A). The most obvious use of the servo-constraint concept could be to prescribe the $r - r^{d}(t) = 0$ or the $\vartheta - \vartheta^{d}(t) = 0$ constraint, where $r^{d}(t)$ or $\vartheta^{d}(t)$ are appropriate functions of time. Let us consider, for example, a horizontal path prescribed by x(p) = p and y(p) = -a. Now, the parameter p can be eliminated from (5.1) to obtain the servo-constraint equation for the generalized coordinates of the robot in the form:

$$\left[r^2 - a^2 (1 + \tan^2 \vartheta) \right] = \left[0 \right].$$
(5.2)

Note that (5.1) can be reformulated in the form of servo-constraint in closed form for special x(p) and y(p) only, like the horizontal path above. This is a disadvantage of the servo-constraint concept.

In what follows, we compare the above written task definition approaches A) and B) in general. Consider an underactuated multibody system described by equation of motion

$$\mathbf{M}(\bar{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{H}(\bar{\mathbf{q}})\mathbf{u}, \qquad (5.3)$$

with mass matrix $\mathbf{M}(\bar{\mathbf{q}}) \in \mathbb{R}^{n \times n}$, vector of external forces $\mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \in \mathbb{R}^n$, control input $\mathbf{u} \in \mathbb{R}^l$ and input matrix $\mathbf{H}(\bar{\mathbf{q}}) \in \mathbb{R}^{n \times l}$.

5.1.2 Classical servo-constraint

In case A), the desired task can only be fulfilled if the dimension of the servo-constraint $\boldsymbol{\sigma}(\mathbf{\bar{q}}, t) \in \mathbb{R}^{l}$ equals to the number of control inputs l. As in the case of the method of Lagrange multipliers, the servo-constraint equation $\boldsymbol{\sigma}(\mathbf{\bar{q}}, t) = \mathbf{0}$ can be written on the level of accelerations as equation (3.31) shows. This way, the control input \mathbf{u} can be calculated from

$$\begin{bmatrix} \mathbf{M}(\bar{\mathbf{q}}) & -\mathbf{H}(\bar{\mathbf{q}}) \\ \boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}},t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{u} \end{bmatrix} = \\ \begin{bmatrix} & -\mathbf{C}(\bar{\mathbf{q}},\dot{\bar{\mathbf{q}}}) \\ -\dot{\boldsymbol{\sigma}}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}},\dot{\bar{\mathbf{q}}},t)\dot{\bar{\mathbf{q}}} - \dot{\boldsymbol{\sigma}}_t(\bar{\mathbf{q}},\dot{\bar{\mathbf{q}}},t) - \mathbf{K}_{\alpha}(\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}},t)\dot{\bar{\mathbf{q}}} + \boldsymbol{\sigma}_t(\bar{\mathbf{q}},t)) - \mathbf{K}_{\beta}\boldsymbol{\sigma}(\bar{\mathbf{q}},t) \end{bmatrix}, \qquad (5.4)$$

where the acceleration level servo-constraint equation is stabilized by means of the gains \mathbf{K}_{α} and \mathbf{K}_{β} , similarly to the Baumgarte method. The actual calculation can be carried out in the very same way as in the method of Lagrange multipliers detailed in Section 3.3.2 (see (3.78)).

5.1.3 Parametric path

If we use parametric functions for the specification of the end-effector path (case B)), the task is defined by

$$\boldsymbol{\psi}(\bar{\mathbf{q}}, \mathbf{p}, t) = \mathbf{0}, \qquad (5.5)$$

where $\boldsymbol{\psi} \in \mathbb{R}^n$. The time dependent parameters are collected in vector $\mathbf{p} \in \mathbb{R}^{n-l}$. From the second time derivative of (5.5), the acceleration $\ddot{\mathbf{p}}$ and $\ddot{\mathbf{q}}$ can be expressed:

$$\begin{split} \boldsymbol{\psi}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}},\mathbf{p},t)\ddot{\bar{\mathbf{q}}} + \boldsymbol{\psi}_{\mathbf{p}}(\bar{\mathbf{q}},\mathbf{p},t)\ddot{\mathbf{p}} + \\ \dot{\boldsymbol{\psi}}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}},\mathbf{p},\dot{\bar{\mathbf{q}}},\dot{\mathbf{p}},t)\dot{\bar{\mathbf{q}}} + \dot{\boldsymbol{\psi}}_{\mathbf{p}}(\bar{\mathbf{q}},\mathbf{p},\dot{\bar{\mathbf{q}}},\dot{\mathbf{p}},t)\dot{\mathbf{p}} + \dot{\boldsymbol{\psi}}_{t}(\bar{\mathbf{q}},\mathbf{p},\dot{\bar{\mathbf{q}}},\dot{\mathbf{p}},t) = \mathbf{0}\,, \end{split} \tag{5.6}$$

where index t refers to explicit time derivative and the dots refer to total time derivative, like in equations (3.25) and (3.26). The following form can be constructed again:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & -\mathbf{H} \\ \boldsymbol{\psi}_{\bar{\mathbf{q}}} & \boldsymbol{\psi}_{\mathbf{p}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\mathbf{p}} \\ \mathbf{u} \end{bmatrix} = \\ \begin{bmatrix} & -\mathbf{C}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) \\ -\dot{\boldsymbol{\psi}}_{\bar{\mathbf{q}}} \dot{\bar{\mathbf{q}}} - \dot{\boldsymbol{\psi}}_{\mathbf{p}} \dot{\mathbf{p}} - \dot{\boldsymbol{\psi}}_{t} - \mathbf{K}_{\alpha}(\boldsymbol{\psi}_{\bar{\mathbf{q}}} \dot{\bar{\mathbf{q}}} + \boldsymbol{\psi}_{\mathbf{p}} \dot{\mathbf{p}} + \boldsymbol{\psi}_{t}) - \mathbf{K}_{\beta} \boldsymbol{\psi} \end{bmatrix}.$$
(5.7)

While the measured values of $\bar{\mathbf{q}}$ and $\dot{\bar{\mathbf{q}}}$ are subtituted into the right hand side of (5.7), the actual values of \mathbf{p} and $\dot{\mathbf{p}}$ are calculated by numerical integration of $\ddot{\mathbf{p}}$. This way $\ddot{\mathbf{p}}$ and \mathbf{u} are determined by the direct solution of the linear system (5.7) in every time step. The method can be extended to be applicable for systems with geometric constraints [60].

In case of domestic applications the execution time of the task is not a key problem, so task definition concept B) may be appropriate instead of approach A). In future work, the two methods should be compared also from other viewpoints as well, like path tracking accuracy, computation time demands and stability.

5.2 Inverted pendulum-like systems

In this section, we investigate the possibility of stabilizing a floating body that is unstable without parametric excitation. We construct an underactuated mechanical system where the periodic motion caused by paddling appears in the form of parametric excitation at the actuator. By using the stability chart of the Mathieu equation, we can find sets of parameters where the stabilization of the floating body is realizable, that is, the desired vertical position is achieved without an actual actuator along the corresponding DoF. This can help the canoeist in the stabilization of the canoe, but it can also cause stable ships to capsize [61-64].

5.2.1 Stability of row-vessels

Flat-water paddling sports especially kayaking and canoeing are fairly popular water-sports. Paddling sports usually demand very sophisticated and well synchronized body motion, furthermore, balancing is a difficult task in itself for a beginner athlete. Canoes and kayaks are pretty labile, using mechanical terminology, we call them statically unstable. In this chapter, we focus on the balancing of these row-vessels.

Although, it is usually the boat's lengthwise acceleration that is in the centre of interest of coaches, the boat oscillates in vertical direction, too, due to the rowing motion of the athlete. We construct a 3 DoF dynamical model with a servo-constraint that prescribes the centre of gravity to move periodically in the vertical direction. The so-called parametric excitation arises in these systems due to the vertical periodic motion of the mass centre, and the equation of motion is obtained in the form of the so-called Mathieu equation. This unusual class of excitation is studied with respect to the stability of the system composed by the athlete and the row-vessel.

The main objective of this work is to investigate the possibility of the stabilization of a normally unstable floating body by parametric excitation. First, we briefly summarize the theory, since parametric excitation cannot be investigated analytically in closed form even in linear systems. Secondly, by using the so-called Incze-Strutt stability chart of the Mathieu equation, we define sets of parameters where the stabilization of a normally unstable floating rigid body is possible. This has an effect, for example, that can help in stabilization in case of a canoeist. Experimental and numerical investigations were also accomplished to support the idea.

5.2.2 Background of parametrically excited systems

Parametric excitation means that some of the parameters of the system change as a periodic, quasiperiodic or stochastic function of time. Floquet Theory was developed to investigate the stability of these linear systems. The most general periodic form of a parametrically excited system is given by the Floquet-equation [65]:

$$\mathbf{y}'(t) + \mathbf{A}(t)\mathbf{y}(t) = \mathbf{0},\tag{5.8}$$

where matrix **A** is 2π periodic:

$$\mathbf{A}(t+2\pi) = \mathbf{A}(t) \,. \tag{5.9}$$

System (5.8) can be reduced to the Hill equation, which is scalar and typical in mechanical systems due to the presence of the acceleration term:

$$x''(t) + p(t)x(t) = 0, (5.10)$$

where the time dependent parameter p (that could be the stiffness of the systems) is periodic:

$$p(t+2\pi) = p(t).$$
(5.11)

If the coefficient of x in the Hill-equation (5.10) is harmonic, we obtain the so-called Mathieu equation [66], which is the simplest form of a parametrically excited system:

$$x''(t) + (\delta + \varepsilon \cos(t)) x(t) = 0.$$
(5.12)

The Mathieu-equation can be transformed to Floquet-equation form, when the periodic coefficient matrix is written as:

$$\mathbf{A}(t) = \begin{bmatrix} 0 & 1\\ -\delta - \varepsilon \cos(t) & 0 \end{bmatrix}.$$
 (5.13)

Parametric excitation is usually considered as an unexpected cause of instability problems, but under certain conditions, it can also be used for stabilizing unstable processes or equilibria. The main idea is to eliminate an oscillation with the help of another oscillation. The oldest known example is the inverted pendulum which can be stabilized by the harmonic vibration of the suspension pivot point [67, 68]. We examine whether it is possible to stabilize a normally unstable floating rigid body in a similar way, and if so, for what ranges of parameters.

The stability chart of the Mathieu equation was derived in 1928, see references [69, 70]. It is shown in Fig. 5.2 and known as Incze-Strutt diagram.



FIGURE 5.2. Ince-Strutt diagram: stability chart of Mathieu-equation

The basic idea was to find the stability boundaries in a double series expansion with respect to the parameter δ and the solution x as a function of the "small" parameter ε that is the amplitude of the parametric excitation:

$$\delta(\varepsilon) = \delta_0 + \delta_1 \varepsilon + \delta_2 \varepsilon^2 + \dots, \qquad (5.14)$$

$$x(t,\varepsilon) = u_0(t) + u_1(t)\varepsilon + u_2(t)\varepsilon^2 + \dots$$
(5.15)

We are interested in the $\delta < 0$ region of the Incze-Strutt diagram, were the equilibrium is obviously unstable for $\varepsilon = 0$ but it can become stable for a narrow $\varepsilon > 0$ region. After the application of the Floquet theory, the truncated stability boundaries in question appear in the following form:

$$\delta_1(\varepsilon) = -\frac{1}{2}\varepsilon^2, \qquad (5.16)$$

$$\delta_2(\varepsilon) = \frac{1}{4} - \frac{1}{2}\varepsilon - \frac{1}{8}\varepsilon^2.$$
(5.17)

5.2.3 Stability of the parametrically excited floating body

A floating body, like a ship, has 6 degrees of freedom (DoF), see Fig. 5.3a. The 3 rotations are the roll, pitch and yaw and the transversal motions are the surge, sway and heave. In order to use as simple model as possible for the analytic calculations, we investigate planar motion only (see the vertical plane in Fig. 5.3a). The generalized coordinates of the planar model are chosen to be the roll angle θ and the position coordinates x (sway) and y (heave) of the centre of gravity. The mechanical and geometric parameters shown in Fig. 5.3b are water density ρ , mass m of the vessel, moment of inertia $J_{\rm C}$, length l of the body in direction z (perpendicular to the plane), width a of the body and height p of the mass centre. For the sake of convenience, the height of the wall of the vessel is supposed to be infinitely high. Introduce the modified density parameter μ , with which the shallow dive h can be calculated as follows:

$$\mu = \rho a l \,, \tag{5.18}$$

$$h = \frac{m}{\mu}.$$
(5.19)

The water surface is assumed to be ideally flat and steady. The potential function of the system can be seen in Fig. 5.4a and 5.4b. Rectangular and triangular regions (denoted by R and T) are distinguished. If the roll angle is large, only one corner of the body is in the water, thus the wetted part of the body is triangular shaped. The critical value θ_{cr} separates the two regions.



FIGURE 5.3. Spatial ship motions (a), planar mechanical model of the floating body (b)

Clearly, the vertical position of a symmetric floating body is an equilibrium position. First, we identify other possible equilibria of the square shaped floating body. The stability of each equilibrium is examined by the analysis of the potential function of these conservative systems. The vertical position is stable in Fig. 5.4a and unstable in Fig. 5.4b. The stability loss of the vertical position leads to two tilted equilibrium positions of the body.



FIGURE 5.4. Potencial function of a vertically stable case (a), a vertically unstable case, when two stable equilibria arise (b)

The Lagrangian is given by the difference of the kinetic and the potential energy:

$$L = \frac{m}{2} \left(\dot{x}^2 + \dot{y}^2 \right) + \frac{J_{\rm C}}{2} \dot{\theta}^2 - mgy - \mu g \left(\frac{p^2}{2} \cos \theta - py + \frac{y^2}{2 \cos \theta} + \frac{a^2}{24} \tan \theta \sin \theta \right).$$
(5.20)

Since the damping effect of the water is neglected, the corresponding Lagrangian equations of the second kind can be arranged in the form

$$m\ddot{x} = 0, \qquad (5.21)$$

$$m\ddot{y} + \frac{\mu g}{\cos\theta}y = (\mu p - m)g + u, \qquad (5.22)$$

$$J_{\rm C}\ddot{\theta} + \frac{\mu g}{24} \left(\left(a^2 - 12p^2\right)\sin\theta + \left(a^2 + 12y^2\right)\frac{\tan\theta}{\cos\theta} \right) = 0, \qquad (5.23)$$

for the minimum set of generalized coordinates x, y and θ , where the control input u represents the control force provided by the athlete that makes the centre of gravity moving in the vertical direction relative to the water surface. The equations of motion (5.21)-(5.23) are subjected to the servo-constraint

$$y(t) - y_0 + y_1 \cos(\omega t) = 0, \qquad (5.24)$$

which prescribes the vertical, harmonic oscillation of the centre of gravity of the model composed by the athlete and the vessel. Since the vibration of the mass centre is measured from the vertical equilibrium position, the constant term y_0 in the servo-constraint (5.24) is determined as a function of the roll angle:

$$y_0 = (p-h)\cos\theta. \tag{5.25}$$

Thus, the final form of the servo-constraint is

$$y(t) - (p-h)\cos\theta + y_1\cos(\omega t) = 0,$$
 (5.26)

which is the same as the general form defined in (3.3).

System (5.21)-(5.23) and (5.26) is a 3 DoF underactuated system with one actuator and one task. The actuator force is provided by the athlete's body and contributes to satisfy the servo-constraint (5.26). At the same time, we expect that the same actuator force helps to stabilize the otherwise unstable $\theta = 0$ vertical equilibrium position via parametric excitation.

In order to obtain the system in the form of the Mathieu equation (5.12), we reduce the number of DoF. The generalized coordinate x does not appear in the Lagrangian, so x is a so-called cyclic coordinate. This means that there is no actuator that reaches this DoF, but there is no prescribed task for that either. Therefore we decouple (5.21) from the system. Equation (5.22) can also be separated from the system, and can be used for the calculation of the required control action u. We suppose that the servo-constraint (5.26) is completely satisfied, and the value y defined by the servoconstraint is substituted back into the equation of motion (5.23). Since both x and y are eliminated, the only generalized coordinate left is the roll angle θ , for which the Lagrangian equation of motion reads as:

$$J_{\rm C}\ddot{\theta} + mg\left(\frac{1}{2}\frac{m}{\mu} - p\right)\sin\theta + \mu g\left[\frac{y_1^2\cos^2(\omega t)\tan\theta}{2\cos\theta} + \frac{a^2}{24}\left(\sin\theta + \frac{\tan\theta}{\cos\theta}\right)\right] = 0.$$
(5.27)

We investigate the stability of the vertical position $\theta = 0$, which is affected by the oscillation amplitude y_1 and the angular frequency ω . The equation of motion (5.27) can be linearized. We introduce the dimensionless time $\tau = \omega t$. Finally the linearized equation of motion (5.28) is in complete correspondence with the Mathieu-equation (5.12):

$$\theta'' + (\delta + \varepsilon \cos(\tau)) \theta = 0, \qquad (5.28)$$

where the parameters are:

$$\delta = \frac{\mu g}{4\omega^2 J_{\rm C}} \left[\frac{m}{\mu} \left(\frac{1}{2} \frac{m}{\mu} - p \right) + \frac{y_1^2}{4} + \frac{a^2}{12} \right], \qquad (5.29)$$

$$\varepsilon = \frac{\mu g}{4\omega^2 J_{\rm C}} \frac{y_1^2}{4} \,. \tag{5.30}$$

If $\varepsilon = 0$ and $\delta < 0$ then the vertical position is unstable and the boat capsizes, but it can be stable if the amplitude y_1 increases, that is, when $\varepsilon \neq 0$.

We can apply the Incze-Strutt diagram (see [69] and Fig. 5.2) for the equation (5.28) of motion of the floating body with the physical parameters (5.29) and (5.30). The transformed stability chart in Fig. 5.5 shows the physical parameters of the excitation: the angular frequency ω and the amplitude y_1 . The stability boundaries $y_{1,1}(\omega)$ and $y_{1,2}(\omega)$ are obtained by substituting expressions (5.29) and (5.30) into the equations of the boundaries $\delta_1(\varepsilon)$ and $\delta_2(\varepsilon)$ ((5.16) and (5.17)), respectively. Appropriate finite values of the amplitude and the frequency of the harmonic oscillation are needed for the stabilization of the normally unstable floating body. The stability boundaries are highly sensitive to the changes in the geometrical parameters and the mass. The parameters collected in Table 5.1 are used. The stability boundaries $y_{1,1}(\omega)$ and $y_{1,2}(\omega)$ can be estimated in a conservative way by the asymptotes $\tilde{y}_{1,1}$ and $\tilde{y}_{1,2}(\omega)$ respectively:

$$\tilde{y}_{1,1} = \sqrt{\frac{12\mu mp - 6m^2 - a^2\mu^2}{3\mu^2}},$$
(5.31)

$$\tilde{y}_{1,2}(\omega) = 4\omega \sqrt{(\sqrt{38} - 6)\frac{J_{\rm C}}{\mu g}}.$$
(5.32)

By means of the intersection point ω^* , y_1^* of the asymptotes, an optimal (minimum) value of stroke frequency can be calculated:

$$\omega > \omega^* \approx \frac{1}{4} \sqrt{\frac{g}{h}} \sqrt{\frac{m}{J_{\rm C}} \frac{6h(2p-h) - a^2}{3(\sqrt{38} - 6)}}, \qquad (5.33)$$

where the boat can be stabilized with the minimal oscillation amplitude $\tilde{y}_{1,1}$.

$J_{\rm C}$	m	l	a	p	ω
$2\mathrm{kg}\mathrm{m}^2$	$88 \mathrm{kg}$	$3.6\mathrm{m}$	$0.5\mathrm{m}$	$0.47\mathrm{m}$	$6.28 \mathrm{rad/s}$

TABLE 5.1. Parameters used for stability investigation

5.2.4 Verification by numerical simulation and experiment

This subsection describes a numerical case study. The parameters are taken from a realistic scenario in sports. Athletes are told to keep a good rhythm of rowing and not to row faster but row more powerfully when they want to go faster. Frequency of 80-85 strokes per minute is typical and usually, accelerations of 2 m/s^2 and 150 N force can be measured in horizontal direction [71]. The specific literature rarely refers to the vertical displacements, accelerations and forces, which, in our view, are important in the stabilization process. The forces acting between the shell and the rower and between the water and paddle blade change the resting waterline causing oscillations of 4 - 6 cm [72].



FIGURE 5.5. Stability chart of the parametrically excited floating body with the ω^{e} and y_{1}^{e} values taken from real life experience

These typically experienced rythm and oscillation values are shown by ω^{e} , y_{1}^{e} in Fig. 5.5 just at the boundary of the stable region. The chosen parameters of the numerical simulation are summarized in Table 5.1 with which equations (5.31) and (5.33) give minimal rythm of rowing $\omega^{*} = 3.4 \text{ rad/s}$ and oscillation amplitude $y_{1}^{*} = 58 \text{ mm}$ which are also denoted in Fig. 5.5.

The nonlinear equation of motion (5.28) was used for the numerical simulation. A constant angular frequency has been set, and the stability boundaries were crossed by increasing the amplitude parameter y_1 . We investigate the time history of the roll angle θ and the Poincaré section (see Fig. 5.6). The results confirm perfectly the predicted linear stability limits and they also provide information about kind of vibrations that occur before and after the loss of stability as the stroke amplitude y_1 increases. Note, that the vibrations seem to be chaotic for large excitation amplitudes, which means that while the vessel vibrates with very large amplitudes in the range of 60 degrees, it still does not capsize.

An experiment verified the practical validity of the mechanical model and its analytical and numerical study. A small boat was constructed and two eccentric counter-rotating rotors provided the necessary parametric excitation in the vertical direction (see Fig. 5.7). The physical parameters are collected in Table 5.2. The amplitude of the vertical oscillation caused by the excenter was $y_1 = 15$ mm, and the angular frequency was set to $\omega = 12.56$ rad/s, which is 2 Hz. The experiment clearly showed that the ship stabilization is possible via parametric excitation: the boat floated stably with the rotors running, while it capsized immediately when the rotors were stopped. Realistic and finite parameter domains were found where sportsmen can stabilize their boats with this good rythm via parametric excitation.



FIGURE 5.6. Numerical results: amplitude is too small for the stabilization (a), stable case (b), amplitude is too large for the stabilization (c)



FIGURE 5.7. Experimental tool with stopped rotors (a) and running rotors (b)

J _C	m	l	a	p	ω	y_1
$4.2 \cdot 10^{-4} \mathrm{kg}\mathrm{m}^2$	0.4 kg	$0.27\mathrm{m}$	$0.082\mathrm{m}$	$0.043\mathrm{m}$	$12.56\mathrm{rad/s}$	$0.015\mathrm{m}$

TABLE 5.2. Parameters used for stability investigation

5.3 New results

Thesis 4.

It was proven that the unstable equilibrium position of a floating body on the surface of liquid can be stabilized by means of parametric excitation. The row-vessels, like kayak and canoe, having unstable prescribed equilibrium position, were modelled as underactuated mechanical systems where the athlete can stabilize the vessel only with the periodic vertical motion of his/her centre of gravity, while the vessel is considered as a 3 DoF body moving in the vertical plane. The stability chart of the desired vertical position of the vessel was constructed. By means of this chart, the required rhythm ω of rowing was determined, where the stabilization can already be achived with minimum effort:

$$\omega^* \approx \frac{1}{4} \sqrt{\frac{g}{h}} \sqrt{\frac{m}{J_{\rm C}}} \frac{6h(2p-h) - a^2}{3(\sqrt{38} - 6)} \,,$$

where a is the characteristic width size of the vessel, p is the height of the common centre of gravity of the vessel and the athlete, h is the diving depth, m is the mass and $J_{\rm C}$ is the mass moment of inertia with respect to roll axis at the mass centre. The mechanical model and the results were validated by means of the empirical geometric data and the rowing rhythm values in the range of 75-90 strokes/minute. The stabilizability of unstable equilibria of floating bodies by the use of parametric excitation was also demonstrated by a small scale test rig.

The numerical study of the nonlinear system indicated that small amplitude chaotic oscillations arise when the stabilization by parametric excitation is unsuccessful in Ljapunov sense. This means that the system can be considered practically stable even in these cases.

Related publications: [61–64]

Thesis 5.

The method of Lagrange multipliers was extended for the computed torque control of underactuated robots in cases when the motion of the robot is restricted to a prescribed hypersurface in ith phase-space, where the hypersurface is given as a function of conveniently chosen optional parameters. The number of these parameters is equal to the difference of the degrees of freedom and the number of actuators.

If the prescribed hypersurface can be expressed in implicit form by means of the elimination of the chosen parameters, the corresponding implicit form was defined as a servo-constraint. The computed torque control method was constructed with the corresponding servo-constraint based algorithm, which reduces the size of the numerical task compared to the existing methods in the literature. The size of the task was reduced by the dimension of the eliminated parameters. The elimination of these parameters might be possible, because the time history of the motion on the hypersurface cannot be prescribed in case of underactuated systems.

Related publication: [59]

Chapter 6

Development of the Acroboter platform

The domestic robot platform Acroboter [12] exploits a novel concept of ceiling based locomotion. Acroboter is an acronym that synthetizes the aim of the European Union 6th Framework Project (IST-2006-045530) which was to develop "Autonomous Collaborative Robots to Swing and Work in Everyday Environment". The project officially and successfully ended, but some developements are going on and supported by other resources.

The robot platform is designed to perform pick and place tasks as well as carry other service robots with lower mobility. The Acroboter platform extends the workspace of these robots to the whole cubic volume of the indoor environment by utilizing the almost obstacle free ceiling.

As the Acroboter is a crane-like robot, first we summarize some basic results about crane dynamics and control. In the subsequent sections we apply the presented control concepts to the Acroboter system, which is complex enough to be handled as an underactuated and redundant multibody system.

6.1 Introduction to dynamics of crane-like systems

We investigate the motion of swinging payloads connected to a cable of moving suspension point by extending the methods of robotics. In general, the suspension point is moved by a crane, but in specific applications like the Acroboter [12], it can also be moved by a robot. The swinging load has to be capable to follow spatial target trajectories and to keep desired positions with adequate accuracy. The main purpose is to solve the inverse kinematical and dynamical problem in order to provide computed torque control (CTC) of this crane structure considered as an underactuated robot. The presented results gained from the algebraic analysis of a simplified spatial model show the opportunities and the limitations of trajectory tracking and control [73].

Cranes are pendulum-like structures that are widely used for transporting a payload to a specified position, which is usually defined accurately. Additionally, the payload sometimes has to follow a prescribed spatial trajectory, too. Since cranes are nonlinear oscillating systems, it is a complicated task to achieve a good motion control which suppresses the swinging motion of the payload. In the general cases of tower crane, gantry crane, overhead crane, floating crane and aerial crane shown in Fig. 6.1, the position control of the payload is realized by the controlled movement of the top mounting point of the cable. The low actuating possibilities also complicate the controlling strategies. The oscillating behavior of floating and aerial cranes significantly intensifies because the position of the top mounting point is also perturbed by environmental effects such as waves and wind.



FIGURE 6.1. tower crane (a), gantry crane (b), overhead crane (c), floating crane (d) and aerial crane (e)

In order to achieve high efficiency, the automation of the crane movements is significant. Several controlling strategies have been developed; most commonly feedback controllers [5] and time delayed feedback controllers are used for oscillation suppression [4, 5, 74, 75]. In order to attain better antisway control, fuzzy logic controllers [5, 76] and genetic algorithm-trained neuro-controllers are also used [77].

The complexity of the inverse dynamical calculation comes from the dynamics of the pendulumlike system. A rigid armed structure like a robotic arm can be actuated directly by the servomotors built in the joints. Contrarily, the actuation applied on the top/mounting point of the cable does not have direct effect on the swinging load. There are no actuators to vary the swinging angles, thus these crane systems are underactuated.

During motion, the cable is not in the vertical hanging-down position. Since the cable can be winded up and down, a certain desired position (x^d, y^d) can be reached with different configurations with different x_m trolley positions as shown in Fig. 6.2a, which means that beside the underactuated nature of the system, kinematic redundancy is also present.



FIGURE 6.2. Redundancy of the varied length pendulum (a), simplified model of a crane (b)

We examine a simplified crane model shown in Fig. 6.2b with analytic calculations and we investigate the inverse kinematics and dynamics of the system. The solution of the inverse dynamical problem can be given by analytic expressions. Since the method of computed torques is not robust

in itself, we apply a feedback controller additionally to the analitical inverse dynamical formulae. The mounting point is moved in the z = 0 horizontal plane by a two degree of freedom mechanical structure which can be an RR, RT (tower crane) or a TT (for example overhead crane) structure. To this end, the mounting point is attached to a simple block with mass $m_{\rm m}$ placed in the horizontal plane, which can be moved by any type of 2 DoF planar mechanism. The mass of the payload is m. The cable interconnecting the mounting point and the payload is assumed to be massless without any bending stiffness.

6.1.1 Lagrangian generalized coordinates and singularities

The minimum set of generalized coordinates of the spatial mechanical model (see Fig. 6.2b) is chosen as:

$$\bar{\mathbf{q}} = \left[\vartheta \ \psi \ l \ x_{\mathrm{m}} \ y_{\mathrm{m}}\right]^{\mathrm{T}} , \qquad (6.1)$$

where ϑ is the nutation angle of the cable, the precession angle is denoted by ψ , l is the cable length that can be varied by the winding mechanism. The position of the mounting point is given by the planar coordinates $x_{\rm m}$ and $y_{\rm m}$. Thus, the load as a point-mass has 3 DoF, the cable mounting point has another 2 DoF, so the system has overall 5 DoF. Using the minimum set of generalized coordinates (6.1), the mass matrix reads

$$\mathbf{M}(\mathbf{q}) = \begin{vmatrix} ml & 0 & 0 & m\vartheta_{c}\psi_{c} & m\vartheta_{c}\psi_{s} \\ 0 & ml\vartheta_{s}^{2} & 0 & -m\vartheta_{s}\psi_{s} & m\vartheta_{s}\psi_{c} \\ 0 & 0 & m & m\vartheta_{s}\psi_{c} & m\vartheta_{s}\psi_{s} \\ m\vartheta_{c}\psi_{c} & -m\vartheta_{s}\psi_{s} & m\vartheta_{s}\psi_{c} & m+m_{m} & 0 \\ m\vartheta_{c}\psi_{s} & m\vartheta_{s}\psi_{c} & m\vartheta_{s}\psi_{s} & 0 & m+m_{m} \end{vmatrix} , \qquad (6.2)$$

where subscript c and s refers to sine and cosine functions, respectively.

The equations of motion are singular at the hanging down position, where the nutation angle is zero, that is, the mass matrix is non-invertible if $\vartheta = 0$ [73]. Since the motion at and around the hanging down position is significant in this analysis, we need another approach to construct the equations of motion, which is detailed in the next subsection.

6.1.2 Cartesian descriptor coordinates

We choose the Cartesian coordinates of the mass centre of the load x, y and z as generalized coordinates instead of using ϑ , ψ and l. So the new vector of the generalized coordinates is:

$$\bar{\mathbf{q}} = \begin{bmatrix} x \ y \ z \ x_{\mathrm{m}} \ y_{\mathrm{m}} \end{bmatrix}^{\mathrm{T}} . \tag{6.3}$$

Note that this set of coordinates is still minimum number. A constant mass matrix is obtained:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 \\ 0 & 0 & 0 & m_{\mathrm{m}} & 0 \\ 0 & 0 & 0 & 0 & m_{\mathrm{m}} \end{bmatrix},$$
(6.4)

where we can see the advantage of using natural coordinates: the structure of the mass matrix is much sipler and does not depend on intricate trigonometric functions of descriptor coordinates.

The above introduced models of cranes will be used in Section 6.3 for the simplest possible mechanical model of the Acroboter platform, but first the complete structure of the Acroboter is discussed.

6.2 Structure of Acroboter

Recently, more and more robotic systems try to utilize the advantages of the underactuation, like agile motion and energy efficient operation. Besides, a new direction in the development of indoor service robots is the use of robotic structures that can move on the almost obstacle free ceiling of a room, while transport the payload or an actuator mechanism similarly to gantry cranes. The conceptual design and the prototype of the ceiling based underactuated robot called Acroboter [12, 47, 78] are shown in Fig. 6.3 and 6.4, respectively. Similar concepts, for example, are the Flora service robot [79] that utilizes permanent magnets to keep and move its mobile cart on the ceiling and a tethered aerial robot presented in [80], in which the working unit is suspended on a single cable and equipped with two ducted fan actuators.



FIGURE 6.3. Conceptual design of the Acroboter system

Obstacle avoidance is an important problem in service and mobile robotics, especially when the robots have to move in an everyday indoor environment. The Acroboter utilizes the ceiling that is almost obstacle-free compared to the ground, since it avoids all problems of floor based locomotion caused by randomly placed small objects obstructing the free motion. A grid of anchor points is equipped on the ceiling. A climber unit (CU) moves on the anchor point system by grasping two anchors at the same time or rotating around one anchor. The CU carries a swinging unit (SU) with a system of suspending and orienting cables. The main suspension cable is not only responsible for supporting the SU but provides information signal and power transmission, too. The cable connector unit (CC) connects the main cable to the additional windable secondary cables, which are responsible

for the orientation of the SU and form a parallel kinematic structure. The SU carries the end-effector of the robot which can be a grasper, but it is also possible to interchange the grasper unit to other machines, such as a vacuum cleaner, with the help of the mechanical and electrical interface situated at the bottom of the SU. This interface plays the role of a toolchanger. Since the SU is the working unit of the robot, the objective of the robot is the fine positioning of the SU that accomplishes path following or pick and place tasks. Its motion is controlled by ducted fan actuators additionally to the variable length suspending cables.



FIGURE 6.4. Prototype of the Acroboter system

The CU is a fully actuated planar RRT robot and is able to move the upper mounting point of the main cable in arbitrary location in a plain parallel to the ceiling. To achive this planar motion, 2 actuated revolute joints and 1 linear joint operate on the CU. The 3 DoF of the CU is required because it can move the top mounting point of the main cable even in the case, when the upper part of the CU grasps two anchors at the same time. The length of the main cable and the secondary cables are varied by servo motors, which means that the system has further 4 actuators. The positioning of the SU is assisted by six ducted fan actuators, which can provide a resultant torque around the vertical axis and an arbitrary direction resultant force in the horizontal plane. Consequently, the ducted fan actuatators represent 3 independent control inputs. Totally, the number of actuators is 10. Despite the large number of actuators, the system is still underactuated, because the number of actuators is lower than the total number of DoF, which is 12 beacuse the CU as an RRT robot represents 3 DoF, the cable connector, wich is modeled as a particle, represents another 3 DoF and the SU as a rigid body in space represents 6 DoF [47, 78, 81].

The task of the robot is to keep the SU in the desired position and orientation in space, while there are no requirements for the motion of the CU and the CC. This means that the dimension of the task is only 6, all related to the SU only. Since the dimension of the prescribed task is smaller than the DoF, the robot is kinematically redundant. Thus, the Acroboter can be considered as an underactuated and redundant robot, but this issue will be discussed accurately in details in Section 6.6.

As a summary of the conceptual design of the robot, we can say that the Acroboter concept utilizes the advantages of flying robots: the problem of obstacle avoidance is resolved. Besides, due to the suspending cables, Acroboter needs much less energy than flying robots, because there is no need for lifting force provided by ducted fans. With the help of the crane-like structure, the Acroboter is able to utilize the pendulum-like motion efficiently. This suspended structure provides large vertical workspace while it is still lightweight. On the other hand, the limited horizontal actuation on the suspended payload having non-linear oscillations might be critical in certain applications.

6.3 Analytical inverse dynamics of pendulum-like models

In some simple and/or specific cases, analytical solutions may exist for establishing CTC of underactuated systems. The analytical inverse dynamical calculation of an underactuated crane model will be presented in this section. The application of Laplace-transformation provides a general method for linear systems only, but the equations (3.1), (3.2) and (3.3) form a non-linear system. It will be shown that by means of some reasonable simplifications, the complex problem of the inverse dynamical calculation of the Acroboter can be converted into the inverse dynamical problem of a multiple pendulum system. The method will be presented for the spatial double pendulum, and the results will be compared in details with the planar model of Acroboter [14, 82]. The pendulum models in question are presented in Fig. 6.5.

6.3.1 Single pendulum model

The ducted fan actuators generate typically low thrusts compared to the possible total weight of the robot including the payload. Thus, the ducted fan actuators can be excluded from a simplified dynamic model [14, 82] during the inverse dynamical calculation and they are only employed in the linear compensator (feedback control) to eliminate additional small disturbances. The prescribed trajectories of the CU and the windable cables have to be determined to provide the desired pendulum-like motion of the platform during which the orientation of the SU is kept constant (e.g., horizontal) by the secondary cables (see Fig. 6.4). In the control of the system, this trajectory can be used for calculating feedforward torques for the corresponding actuators, while the ducted fans can be used for stabilizing the motion of the swinging unit along its prescribed trajectory. In addition, since the CC has much lower weight than the SU, the CC might be seen as a source of high-frequency small disturbances, only.

These considerations lead to the single pendulum model presented in Fig. 6.5a, where the point mass, $m_{\rm su}$ stands for the swinging unit. The inputs of the inverse calculation are the desired positions $x^{\rm d}(t)$, $y^{\rm d}(t)$ and $z^{\rm d}(t)$ of the centre of mass of the swinging unit.

The goal of the inverse dynamics calculation is to define the corresponding desired trajectory of the CU denoted by x_{cu}^{d} and y_{cu}^{d} , the desired forces which formulates the control input vector: $\mathbf{u} = \left[F_x^d, F_y^d, F_c^d\right]^T$, and the nominal cable length l^d in closed form. Forces F_x and F_y act on the top mounting point of the cable and F_c is the cable force provided by the cable winch motor. The matrix



FIGURE 6.5. Single pendulum (a) and double pendulum (b) models of the Acroboter platform

equation of motion of the single pendulum model can be expressed with minimum set of generalized coordinates $\bar{\mathbf{q}} = [x_{cu} \ y_{cu} \ x \ y \ z]^{T}$ in the form:

where the cable length is expressed as

$$l = \sqrt{(x_{\rm cu} - x)^2 + (y_{\rm cu} - y)^2 + z^2}.$$
(6.6)

The task is defined by the servo-constraint vector (see (3.8)) with $\mathbf{h}(\mathbf{q}) = [x \ y \ z]^{\mathrm{T}}$ and $\mathbf{p}(t) = [x^{\mathrm{d}}(t) \ y^{\mathrm{d}}(t) \ z^{\mathrm{d}}(t)]^{\mathrm{T}}$. The dimension of the servo-constraint vector is 3 since the number of control inputs is also 3 as well as the dimension of the task. Thus, the servo-constraint can be formulated as:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \left[x - x^{\mathrm{d}}(t) \ y - y^{\mathrm{d}}(t) \ z - z^{\mathrm{d}}(t) \right]^{\mathrm{T}}.$$
(6.7)

Here, the separation of controlled and uncontrolled coordinates is unique and trivial (see (3.12)):

$$\mathbf{q}_{c} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}, \tag{6.8}$$

$$\mathbf{q}_{\mathrm{u}} = \left[\begin{array}{cc} x_{\mathrm{cu}} & x_{\mathrm{cu}} \end{array} \right]^{\mathrm{I}}. \tag{6.9}$$

The servo-constraints can easily be solved for the desired value of the controlled coordinates $x = x^{d}(t)$, $y = y^{d}(t)$ and $z = z^{d}(t)$. From the equations of motion (6.5), we can calculate the desired or in other words, the nominal values of the uncontrolled coordinates x^{d}_{cu} and y^{d}_{cu} , which gives the

desired trajectory of the climber unit and the control forces F_x^d , F_y^d and F_c^d as follows:

$$x_{\rm cu}^{\rm d} = x^{\rm d} - \frac{\ddot{x}^{\rm d} z^{\rm d}}{\ddot{z}^{\rm d} + g},$$
 (6.10)

$$y_{\rm cu}^{\rm d} = y^{\rm d} - \frac{\ddot{y}^{\rm d} z^{\rm d}}{\ddot{z}^{\rm d} + g},$$
 (6.11)

$$F_x^{\mathbf{d}} = m_{\mathrm{su}} \ddot{x}^{\mathbf{d}},\tag{6.12}$$

$$F_y^{\rm d} = m_{\rm su} \ddot{y}^{\rm d}, \tag{6.13}$$

$$F_{\rm c}^{\rm d} = -m_{\rm su} l^{\rm d} \frac{\ddot{z}^{\rm d} + g}{z^{\rm d}}.$$
 (6.14)

With the above formulae (6.10) and (6.11) it is straightforward to calculate the desired length of the main suspending cable from equation (6.6), and the corresponding winding torque can also be expressed from (6.14) by knowing the dynamic characteristics of the cable winch motor.

An important result of the above calculation is that the use of the single pendulum model in trajectory generation of the CU requires C⁴ continuity of the desired SU trajectory. This is because the CU trajectories need to be two times continuously differentiable to have smooth desired accelerations. It is because that the relative degree with respect to the input-output pairs of control forces and the SU position is r = 4 in case of single pendulum – and it will be r = 6 in case of the double pendulum (for the notion of relative degree, see Section 2.3).

6.3.2 Double pendulum model

A further step in modeling the Acroboter platform could be the double pendulum model presented in Fig. 6.5b [14, 82]. Beyond enabling a more accurate calculation of the desired trajectory of the CU, this model makes it possible to consider the redundant actuation of the windable cables in the vertical direction and also the motion of the CC described by two uncontrolled coordinates. The vertical distance $z_{cc} - z$ between the SU and the CC is considered as a desired parameter that can be obtained as a fixed optimal value from experience.

In the case of the double pendulum model, the control input vector contains the lateral forces acting on the top mounting point of the upper pendulum and the two cable forces that vary the cable lengths l_1 and l_2 . Thus the control input is $\mathbf{u} = \begin{bmatrix} F_x^d & F_y^d & F_{c1}^d & F_{c2}^d \end{bmatrix}^T$. The equation of motion is written with the minimum set of coordinates $\bar{\mathbf{q}} = \begin{bmatrix} x_{cu} & y_{cu} & x_{cc} & y_{cc} & z_{cc} & x & y & z \end{bmatrix}^T$ in the following form:

$$\begin{bmatrix} 0\\0\\m_{cc}\ddot{x}_{cc}\\m_{cc}\ddot{y}_{cc}\\m_{cc}\ddot{z}_{cc}\\m_{cc}\ddot{z}_{cc}\\m_{su}\ddot{x}\\m_{su}\ddot{x}\\m_{su}\ddot{x}\\m_{su}\ddot{z}\end{bmatrix} + \begin{bmatrix} 0\\0\\0\\m_{cc}g\\0\\m_{sc}g\\0\\0\\m_{su}g\end{bmatrix} = \begin{bmatrix} F_x + F_{c1}(x_{cc} - x_{cu})/l_1\\F_y + F_{c1}(y_{cc} - y_{cu})/l_1\\-F_{c1}(x_{cc} - x_{cu})/l_1 + F_{c2}(x - x_{cc})/l_2\\-F_{c1}(y_{cc} - y_{cu})/l_1 + F_{c2}(y - y_{cc})/l_2\\-F_{c1}z_{cc}/l_1 + F_{c2}(z - z_{cc})/l_2\\-F_{c2}(x - x_{cc})/l_2\\-F_{c2}(y - y_{cc})/l_2\\-F_{c2}(z - z_{cc})/l_2\end{bmatrix}.$$
(6.15)

The trajectory of the SU and the vertical position of the CC are prescribed. Thus, the desired motion of the system is defined by the 4 dimensional servo-constraint with $\mathbf{h}(\mathbf{q}) = [x \ y \ z \ z_{cc} - z]^{\mathrm{T}}$ and $\mathbf{p}(t) = [x^{\mathrm{d}}(t) \ y^{\mathrm{d}}(t) \ z^{\mathrm{d}}(t) \ h^{\mathrm{d}}_{\mathrm{cc}}(t)]^{\mathrm{T}}$ where $h^{\mathrm{d}}_{\mathrm{cc}}(t)$ is the desired elevation of the CC above the SU.

The dimension of the servo-constraint vector is also 4:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \left[x - x^{d}(t) \ y - y^{d}(t) \ z - z^{d}(t) \ z_{cc} - z - h_{cc}^{d}(t) \right]^{\mathrm{T}}.$$
(6.16)

Afterwards, the inverse kinematics and dynamics of the double pendulum model can be solved in two steps. First, the SU can be considered as a single pendulum attached to the CC which is interpreted as a floating suspension point above the SU at a desired elevation. Then the resulting trajectory of the cable connector plays the role of the desired trajectory of the bob of a single pendulum system composed of the CC and the CU. Consequently, the desired trajectories of the CC can be expressed similarly to equations (6.10) and (6.11) extended by the corresponding term of the servo-constraint:

$$x_{\rm cc}^{\rm d} = x^{\rm d} + \frac{\ddot{x}^{\rm d} h_{\rm cc}^{\rm d}}{\ddot{z}^{\rm d} + g},$$
 (6.17)

$$y_{\rm cc}^{\rm d} = y^{\rm d} + \frac{\ddot{y}^{\rm d} h_{\rm cc}^{\rm d}}{\ddot{z}^{\rm d} + g},$$
 (6.18)

$$z_{\rm cc}^{\rm d} = z^{\rm d} + h_{\rm cc}^{\rm d}.\tag{6.19}$$

The desired motion of the swinging unit is provided by the force delivered through its suspending cable:

$$F_{\rm c2}^{\rm d} = \frac{m_{\rm su} l_2^{\rm d} (\ddot{z}^{\rm d} + g)}{h_{\rm cc}^{\rm d}} \,. \tag{6.20}$$

As a second step, the desired trajectory of the climber unit can be calculated as

$$x_{\rm cu}^{\rm d} = x_{\rm cc}^{\rm d} + \frac{F_{\rm c2}^{\rm d}(x^{\rm d} - x_{\rm cc}^{\rm d}) - m_{\rm cc} \, l_2^{\rm d} \, \ddot{x}_{\rm cc}^{\rm d}}{m_{\rm cc} \, l_2^{\rm d} (\ddot{z}^{\rm d} + \ddot{h}_{\rm cc}^{\rm d} + g) - F_{\rm c2}^{\rm d} h_{\rm cc}^{\rm d}} (h_{\rm cc}^{\rm d} + z^{\rm d}), \tag{6.21}$$

$$y_{\rm cu}^{\rm d} = y_{\rm cc}^{\rm d} + \frac{F_{\rm c2}^{\rm d}(y^{\rm d} - y_{\rm cc}^{\rm d}) - m_{\rm cc} \, l_2^{\rm d} \, \ddot{y}_{\rm cc}^{\rm d}}{m_{\rm cc} \, l_2^{\rm d} (\ddot{z}^{\rm d} + \ddot{h}_{\rm cc}^{\rm d} + g) - F_{\rm c2}^{\rm d} h_{\rm cc}^{\rm d}} (h_{\rm cc}^{\rm d} + z^{\rm d}).$$
(6.22)

Finally, the cable force in the upper pendulum is

$$F_{c1}^{d} = -l_{1}^{d} \frac{m_{cc} l_{2}^{d} (\ddot{z}^{d} + \ddot{h}_{cc}^{d} + g) + F_{c2}^{d} h_{cc}^{d}}{l_{2}^{d} (h_{cc}^{d} + z^{d})}, \qquad (6.23)$$

which corresponds to the main cable force.

The results of a numerical example can be seen in Fig. 6.6. The desired trajectory x^{d} of the SU is given by a 6th degree polynomial function that has continuous time derivatives even for the snap, while the desired y trajectory is $y^{d}(t) = 0$ (planar motion) and the desired vertical trajectory is given by the desired values $z^{d}(t) = -1.5$ m and $h_{cc}^{d}(t) = 0.35$ m. The top panel in Fig. 6.6a presents the desired trajectory x^{d} , where the crosses denotes the data used for fitting the polynomial. The bottom left panel summarizes the results of the numerical simulations applied for the same desired trajectory segment.

Significant difference shows up between the desired x_{cu} trajectory calculated by the single or by the double pendulum model. This means that the ducted fan actuators will have to compensate much less errors when the double pendulum model is used.



FIGURE 6.6. Numerical example for the inverse calculation of the planar single pendulum and double pendulum models: time histories (a), comparison of the stroboscopic motion of the controlled single and double pendulum (b)

6.3.3 Multiple pendulum model

Note that the above derivation can be repeated in closed form for the N pendulum model, the dynamics of which is given by the equations

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ m_{k}\ddot{x}_{k} \\ m_{k}\ddot{y}_{k} \\ m_{k}\ddot{y}_{k} \\ m_{k}\ddot{x}_{k} \\ m_{k}\ddot{y}_{k} \\ \vdots \\ m_{k}\ddot{x}_{k} \\ \vdots \\ m_{k}\ddot{x}_{k} \\ \vdots \\ m_{k}\ddot{x}_{k} \\ \vdots \\ m_{k}\ddot{x}_{k} \\ \vdots \\ m_{su}\ddot{x} \\ m_{su}\ddot{y} \\ m_{su}\ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_{k}g \\ \vdots \\ 0 \\ 0 \\ m_{su}g \end{bmatrix} = \begin{bmatrix} F_{x} + F_{c1}(x_{1} - x_{cu})/l_{1} \\ F_{y} + F_{c1}(y_{1} - y_{cu})/l_{1} \\ \vdots \\ -F_{C,k}(x_{k} - x_{k-1})/l_{k} + F_{C,k+1}(x_{k+1} - x_{k})/l_{k+1} \\ -F_{C,k}(x_{k} - x_{k-1})/l_{k} + F_{C,k+1}(x_{k+1} - x_{k})/l_{k+1} \\ -F_{C,k}(x_{k} - x_{k-1})/l_{k} + F_{C,k+1}(x_{k+1} - x_{k})/l_{k+1} \\ \vdots \\ -F_{C,k}(x_{k} - x_{k-1})/l_{k} + F_{C,k+1}(x_{k+1} - x_{k})/l_{k+1} \\ 0 \\ m_{su}g \end{bmatrix} , \quad (6.24)$$

with k = 1, ..., N - 1 representing many possible cable connector points of mass m_k . While this model has no relevance in the CTC design of the Acroboter, the corresponding analytical formulae of the control inputs serve as a reference benchmark problem for testing any CTC algorithms.

6.3.4 Experimental results

Simulation and experimental results [14] corresponding to the single pendulum model of Acroboter are presented in Fig. 6.7b. The required path of the robot (centre of gravity of the swinging unit)

was set to a slanted rectangle with 300 mm by 400 mm projected size onto the horizontal plane and with 130 mm elevation in the z direction. These dimensions correspond to the maximum workspace of the prototype experimental setup, in which the climber unit and the winding actuator of the main suspending cable were emulated by a Hirata MB-H230/MB-H180 3D Cartesian robot (see in Fig. 6.7a). The single pendulum model is characterized by the estimated cable length of 820 mm and the weight 2.8 kg of the swinging unit, while the elevation of the cable connector above the swinging unit was set to 500 mm.



FIGURE 6.7. Experimental setup using an early SU prototype and a cartesian robot instead of CU (a) experimental results based on the single pendulum model: dashed lines show the desired and continuous ones show the measured values (b)

The desired trajectories were calculated by using 6th degree polynomial approximation of the required rectangular path. During experiments the HIRATA robot was commanded to move along the analytically calculated CU trajectories providing an open loop continuous path control for realizing the desired motion of the SU. The position of the swinging unit was measured by a Zebris ultrasonic system. In Fig. 6.7b, the experimental results show that even the applied open loop controller can provide tolerable errors, since the maximum deviation (abs error) from the desired path remained below 60 mm that may be easily compensated by the ducted fans of the swinging unit with a linear compensator.

6.4 Inverse dynamics of planar Acroboter

In tis section, we consider the planar Acroboter only, and we proove that the control of the orientation of the SU by means of the secondary cables can be nested into the analytically determined inverse dynamics of the double pendulum model. This means that the CTC of the planar Acroboter can be designed analytically [14, 82, 83] and this serves as a benchmark problem for the testing of the full CTC control based on backward Euler and Lagrange multiplier methods. The corresponding analytical and numerical results will be compared and this will serve as a basis for the full CTC of the spatial Acroboter.

6.4.1 Analytical approach

In order to present the applicability of the results obtained for the single and double pendulum models, the complete planar model of the Acroboter in Fig. 6.8 is considered which also includes the secondary (orienting) cables of the swinging unit (SU).



FIGURE 6.8. The planar, natural coordinates based Acroboter model for numerical simulations

The dimension of the SU is characterized by the variables a and b, which together with the vertical distance c determine the location of the centre of mass of the SU. In addition, coordinate ξ denotes the relative distance of the cable connector (CC) relative to the left endpoint A of the rod that represents the SU, while h_{cc} is the elevation of the CC above the horizontal rod.

The desired orientation of the swinging unit is considered to be fixed, and without the loss of generality it is assumed that the swinging unit is kept always horizontal by the secondary orienting cables. This restriction is reasonable in a wide range of possible applications, such as pick and place tasks.

The desired horizontal orientation of the SU implies that the corresponding angular acceleration is zero, which yields the equation of motion of the planar model in the simplified form:

$$m\ddot{x} = F_1 \cos\varphi_1 - F_2 \cos\varphi_2,\tag{6.25}$$

$$m\ddot{z} = F_1 \sin\varphi_1 + F_2 \sin\varphi_2 - mg, \tag{6.26}$$

$$0 = -F_1 \left(a \sin \varphi_1 + c \cos \varphi_1 \right) + F_2 \left(b \sin \varphi_2 + c \cos \varphi_2 \right), \tag{6.27}$$

where F_1 and F_2 are the cable forces of the secondary cables and φ_1 and φ_2 are their angles at the SU, respectively.

In equations (6.25)-(6.27) the number of unknown parameters is 5 in total: ξ , F_1 , F_2 , φ_1 and φ_2 since the trajectories x(t) and y(t) of the centre of mass of the SU are prescribed. When the elevation h_{cc} of the cable connector above the SU is also a desired parameter, the necessary independent constraint equations can be formulated for the cable angles as:

$$\cos\varphi_1 = \frac{\xi}{\sqrt{\xi^2 + h_{\rm cc}^2}},\tag{6.28}$$

$$\cos\varphi_2 = \frac{a+b-\xi}{\sqrt{(a+b-\xi)^2 + h_{cc}^2}}.$$
(6.29)

Then the solution of the system of equations (6.25)-(6.29) results in the desired relative position of the cable connector in the form:

$$\xi^{d} = a + \frac{(c + h_{cc}^{d})\ddot{x}^{d}}{\ddot{z}^{d} + g}, \qquad (6.30)$$

which reduces exactly to (6.17) with the substitutions of the expressions

$$x_{\rm cc}^{\rm d} = x^{\rm d} - a + \xi^{\rm d} \,, \tag{6.31}$$

$$z_{\rm cc}^{\rm d} = z^{\rm d} + c + h_{\rm cc}^{\rm d} \,. \tag{6.32}$$

The above heuristic derivation demonstrates that due to the assumption related to the SU horizontality, the complete planar model and the simplified double pendulum model result the same desired trajectory for the CC and consequently, the desired trajectory of the CU is also the same in both models.

6.4.2 Comparison of analytical CTC and backward Euler based CTC

The planar Acroboter model is examined with one of the presented computed torque methods, which is based on the backward Euler discretization of the DAE system (see Section 3.2).



FIGURE 6.9. The planar, natural coordinates based Acroboter model for numerical simulations

The mechanical model is shown in Fig. 6.9. The CU is modeled by a linear drive and its position is denoted by the coordinate x_1 . The CC is modeled by a particle of mass m_{cc} and denoted by P₂ in Fig. 6.9. The SU is modeled by a rigid body of total mass m_{su} and mass moment of inertia J_{CG} , while its descriptor coordinates are the fixtures of the secondary cables denoted by P₃ and P₄, respectively. The center of gravity of the SU is denoted by CG and ρ_{CG} points into its position in the body frame. Summarizing, the system is described by the non-minimum set of descriptor coordinates:

$$\mathbf{q} = \begin{bmatrix} x_1 & x_2 & z_2 & x_3 & z_3 & x_4 & z_4 \end{bmatrix}^{\mathrm{T}} .$$
 (6.33)

The position of the linear drive is controlled by force $F_{\rm L}$ and the control forces acting on the cables of lengths L_1 , L_2 and L_3 are F_1 , F_2 and F_3 , respectively. The control force $F_{\rm T}$ provided by the ducted fan is not used in the CTC algorithm, it will provide an additional linear compensator augmented to the CTC. Thus the control input can be written as:

$$\mathbf{u} = \begin{bmatrix} F_{\mathrm{L}} & F_{1} & F_{2} & F_{3} \end{bmatrix}^{\mathrm{T}}, \qquad (6.34)$$

The mass of the SU is $m_{\rm su} = 5 \,\rm kg$ and that of the CC is $m_{\rm cc} = 0.5 \,\rm kg$. The center of gravity is positioned at the midpoint of the L_{34} line, which is the distance of the cable fixtures P₃ and P₄. The corresponding geometric constraint vector is the following:

$$\varphi(\mathbf{q}) = \left[(x_4 - x_3)^2 + (z_4 - z_3)^2 - L_{34}^2 \right] \,. \tag{6.35}$$

The desired trajectory of the SU is defined as a servo-constraint according to (3.8). The servoconstraint vector $\boldsymbol{\sigma}(\mathbf{q}, \mathbf{p}(t)) = \mathbf{h}(\mathbf{q}) - \mathbf{p}(t)$ is written with:

$$\mathbf{h}(\mathbf{q}) = \begin{bmatrix} \frac{1}{2}(x_3 + x_4) & \frac{1}{2}(z_3 + z_4) & z_2 - \frac{1}{2}(z_3 + z_4) & z_3 - z_4 \end{bmatrix}^{\mathrm{T}},$$
(6.36)

$$\mathbf{p}(t) = \begin{bmatrix} x^{\mathrm{d}}(t) & z^{\mathrm{d}}(t) & h_{\mathrm{cc}}^{\mathrm{d}}(t) & 0 \end{bmatrix}^{\mathrm{T}}.$$
(6.37)

The desired trajectories are given by C⁴ continuous analytical functions, because the largest relative degree of the planar Acroboter is r = 4. The desired horizontal position $x^{d}(t)$ and the desired vertical position $z^{d}(t)$ of the SU and the desired elevation $h^{d}_{cc}(t)$ of the CC above the SU are prescribed by the function with numerical values in SI units:

$$x^{\rm d}(t) = 0.25 \frac{\pi}{2} \arctan\left(2.5(t-6)\right),$$
 (6.38)

$$z^{\rm d}(t) = -1.5 + 0.02t\,,\tag{6.39}$$

$$h_{\rm cc}^{\rm d}(t) = 0.5$$
. (6.40)

The servo-constraint and the geometric constraint together can be solved for the controlled coordinates as follows

$$z_2 = z^{\rm d} + h_{\rm cc}^{\rm d} \,, \tag{6.41}$$

$$x_3 = x^{\rm d} - \frac{1}{2}L_{34}, \qquad (6.42)$$

$$z_3 = z^{\mathrm{d}}, \qquad (6.43)$$

$$z_4 = z^d \tag{6.44}$$

and can be substituted into the control law (3.42) and (3.43). The controlled and uncontrolled coordinates and the corresponding selector matrices are:

$$\mathbf{q}_{c} = \begin{bmatrix} z_{2} & x_{3} & z_{3} & z_{4} \end{bmatrix}^{\mathrm{T}}, \qquad (6.45)$$

$$\mathbf{r}_{c} = \begin{bmatrix} x_{1} & x_{2} & x_{3} \end{bmatrix}^{\mathrm{T}} \qquad (6.46)$$

The desired uncontrolled coordinate values x_1 , x_2 and x_4 , the Lagrange multiplier λ_1 and the control input forces F_L , F_1 , F_2 and F_3 are calculated by the backward Euler based CTC algorithm (3.50-3.53). The unknowns are scaled in order to obtain better conditioned system.

When the algorithm is tested, the measured values are substituted by the results of the simulation of the controlled dynamical system, which is carried out parallel to the CTC algorithm, since the linear compensator within the CTC needs the position and the velocity errors $\mathbf{q}^{d} - \mathbf{q}$ and $\dot{\mathbf{q}}^{d} - \dot{\mathbf{q}}$. The gain matrices of the linear compensator are given as

$$\mathbf{K}_{\mathrm{P}} = P \mathbf{H}_0 \mathbf{\tilde{K}}, \qquad (6.48)$$

$$\mathbf{K}_{\mathrm{D}} = D\mathbf{H}_{0}\mathbf{\ddot{K}} \tag{6.49}$$

with scalars P = 0.25 N/m, D = 0.02 Ns/m and the distributor matrix

$$\tilde{\mathbf{K}} = \begin{bmatrix} 120 & -30 & 0 & 5 & 0 & 5 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}.$$
(6.50)

Matrix $\tilde{\mathbf{K}}$ is tuned by trial and error method and \mathbf{H}_0 is the input matrix evaluated in a characteristic equilibrium configuration of the robot: $x_1 = 0 \text{ m}$, $x_2 = 0 \text{ m}$, $z_2 = -1 \text{ m}$, $x_3 = -\frac{1}{2}L_{34}$, $z_3 = -1.5 \text{ m}$, $x_4 = \frac{1}{2}L_{34}$ and $z_4 = -1.5 \text{ m}$.

The numerical simulations are accomplished with h = 0.01 s time step, which requires that the CTC controller must compute the desired control inputs in each 10 ms sampling period. The tests showed that the on-board computer of the Acroboter prototype is capable of this challenge. Figure 6.10 shows the results of the numerical simulation.

The desired value of the uncontrolled coordinate x_1 is calculated in closed form by the single and the double pendulum model, and x_1 is also calculated numerically. Fig. 6.10a shows that the result gained from the numerical simulation is somewhat closer to the result of the single pendulum model than the result of the double pendulum model.

The control force signals can be seen in Fig. 6.10b. After the initial perturbations the oscillations of the force signals are suppressed. The high acceleration of the SU at t = 6 s also causes oscillations,



FIGURE 6.10. Numerical simulation of the CTC of the planar Acroboter

especially in the $F_{\rm T}$ signal of the linear drive in the CU. The nominal value of the Lagrange multiplier λ_1 is also calculated by the control algorithm and depicted in Fig. 6.10c. The geometric constraint was violated below 10^{-10} m during the simulation.

The dashed curves show the desired path and the continuous curves show the simulated path of the points P_2 , P_3 and P_4 in Fig. 6.10d.

Figure 6.10e shows the desired and the simulated horizontal position of the midpoint of the swinging unit. The desired value is given in $\mathbf{p}(t)$ and the simulated value is given by $\mathbf{h}(\mathbf{q})$. The error history between the desired and the simulated values is depicted in Fig. 6.10f. In the initial configuration, the horizontal position of the swinging unit has an error of 1 cm. As Fig. 6.10f shows, this error is decreased by the linear compensator.

All the errors in Fig. 6.10f, h, i, j present oscillation in time, but the vertical position error signals of the CC and the SU decay faster than the horizontal one and the tilt of the SU. This can be explained by the fact that the cable winding actuators can act on the vertical position directly, while there is no direct actuation on the horizontal position of the swinging unit and the cable connector.

6.5 Inverse dynamics of spatial Acroboter

We use natural (Cartesian) coordinates to describe the configuration of the lower part robot, while the CU is described by minimum set of generalized cordinates. The whole dynamic model is built in the form of (3.1) and (3.2). The cable lengths $l_{\rm mc}$, l_1 , l_2 and l_3 are also introduced as descriptor coordinates, because it makes it possible to use them in the control explicitly. A set of algebraic equations represents the geometric constraints. The schematic drawing of the spatial Acroboter and the chosen descriptor coordinates are shown in Fig. 6.11. The descriptor coordinates are:

$$\mathbf{q} = \begin{bmatrix} \theta_1 \ \theta_2 \ d_3 \ l_{\rm mc} \ l_1 \ l_2 \ l_3 \ x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2 \ x_3 \ y_3 \ z_3 \ x_4 \ y_4 \ z_4 \ x_n \ y_n \ z_n \end{bmatrix}^{\perp} . \tag{6.51}$$



FIGURE 6.11. Sketch of the natural coordinate based spatial model of the Acroboter

Due to its complexity, the motion control of the complete Acroboter system is decomposed into four main steps.

As a zeroth step, a smooth trajectory with C^6 continuity is generated for the SU position and orientation (for explanation see Section 6.3.1 with relative degree 6 for the spatial double pendulum).

The subsequent four steps are carried out at each sampling period. First, the trajectory shown in Fig. 6.12 of the CU (top mounting point of the main cable) is calculated by means of the pendulum (see Section 6.3.1) or the double pendulum (see Section 6.3.2) approximate inverse dynamical model. These trajectories of the SU and the CU are presented in panels a1 - a4 of Fig. 6.13 with a stroboscopic view.

As a second step, the inverse kinematics of the CU is accomplished. Here, redundancy occures due to the fact that only the 2 Cartesian coordinates are prescribed for the end-effector of a 3 DoF robot. The redundancy in the inverse kinematics calculation is resolved with a special algorithm that will be presented in details in Section 6.6.1. This algorithm minimizes the kinetic energy during the motion of the CU in accordance with equations (6.55) and (6.57) explained later. The corresponding stroboscopic figures of RRT manipulator are shown in panels b1 - b4 of Fig. 6.13.

In the third step, the control torques for the cable winches of the main and secondary cables are calculated by a reduced inverse dynamic model only for the CC and the SU. The backward Euler based method (see Section 3.2) is used for the calculation of the control input.

In the forth step, the thrust forces for the ducted fan actuators equipped on the SU are calculated from a simple linear compensator based on the error of the measured SU horizontal coordinates and the measured orientation about the vertical axis relative to the desired ones.

The resulting motion of the SU is depicted with a stroboscopic view in panels c1- c4 of Fig. 6.13. The simulation results indicated that the above described hierarchic CTC algorithm developed for this underactuated redundant robot can be realized with a 100 Hz sampling time on the available processors. The international Acroboter team developed and implemented the corresponding C language program together with all the hardware and drivers, and the Acroboter robot demonstrated its operation in laboratory circumstances [12]. Clearly, the development of the system is still in an immature phase, and several further modifications and improvements can and will be done. Among these, a further development of the control algorithm is mentioned in the subsequent section.



FIGURE 6.12. The results of the inverse calculation of the climber unit




FIGURE 6.13. Simulation results for the CTC of the spatial Acroboter: the sequences of the motion obtained by means of the single pendulum inverse dynamics model are shown in panels a1, a2, a3 and a4; the inverse kinematics of the CU is shown in panels b1, b2, b3 and b4; the obtained motion of the complete Acroboter model is shown in panels c1, c2, c3, and c4

6.6 Generalization for cases of kinematic redundancy

There are many multibody systems, where the number of independent joint coordinates is higher than the minimal number that would be enough to execute the current task. In such cases the system is called *kinematically redundant*. The exact definition of kinematic redundancy is that the dimension of the task is less than the number of DoF. The inverse kinematical calculation, with which the joint coordinates are determined according to the prescribed end-effector poses, is a crucial problem of these robots [15, 16]. This means that we need extra objectives or constraints to solve the inverse kinematical problem. The extra objectives can be chosen optionally. In the followings, the most widespread methods for inverse kinematical calculations are explained for fully actuated systems described by minimum set of generalized coordinates, and after that, the redundancy resolution will be applied for underactuated systems described by non-minimum set of descriptor coordinates.

6.6.1 Redundancy resolution

In general, the main objective is to follow a desired trajectory. It means that the operational space coordinates \mathbf{x} (end-effector coordinates identifying positions and orientations) must be equal to some desired values $\mathbf{x}^{d}(t)$ at every time instant. At the same time, the operational space coordinates can be expressed as an $\mathbf{f}(\bar{\mathbf{q}})$ function of the minimum set of joint/generalized coordinates, therefore we can write:

$$\mathbf{x} = \mathbf{f}(\bar{\mathbf{q}}). \tag{6.52}$$

Using equation (6.52), the task can be given as a servo-constraint (see (3.3)):

$$\boldsymbol{\sigma}(\bar{\mathbf{q}},t) = \mathbf{f}(\bar{\mathbf{q}}) - \mathbf{x}^{\mathrm{d}}(t).$$
(6.53)

Most of the methods use the $\mathbf{f}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}}) = \partial \mathbf{f}(\bar{\mathbf{q}})/\partial \bar{\mathbf{q}}$ Jacobian of $\mathbf{f}(\bar{\mathbf{q}})$ and its null-space to find the optimal motion of the redundant system. When using the term of servo-constraint, the Jacobian $\sigma_{\bar{\mathbf{q}}}(\bar{\mathbf{q}}) = \mathbf{f}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}}) = \partial \sigma(\bar{\mathbf{q}}, t)/\partial \bar{\mathbf{q}}$ is used. The Jacobian matrix transforms uniquely the joint/generalized velocities into end-effector/operation space velocities:

$$\dot{\mathbf{x}} = \boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\dot{\bar{\mathbf{q}}}.$$
(6.54)

If there is no kinematic redundancy, the Jacobian $\sigma_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})$ is a square matrix, and the joint velocity vector $\dot{\bar{\mathbf{q}}}$ can simply be calculated by the inverse of the Jacobian. In the case of kinematically redundant systems, the inverse of the non-square Jacobian matrix does not exist. The pseudo-inverse or the Moore – Penrose generalized inverse of the Jacobian $\sigma_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})$ can be used. The velocity vector $\dot{\bar{\mathbf{q}}}$ calculated from the pseudo-inverse can be complemented with the null-space projection of an arbitrarily chosen joint velocity vector $\dot{\boldsymbol{\xi}}$:

$$\dot{\bar{\mathbf{q}}} = \boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\dagger}(\bar{\mathbf{q}})\dot{\mathbf{x}} + \left(\mathbf{I} - \boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\dagger}(\bar{\mathbf{q}})\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\right)\dot{\boldsymbol{\xi}}.$$
(6.55)

In equation (6.55) the term $\mathbf{I} - \sigma_{\bar{\mathbf{q}}}^{\dagger}(\bar{\mathbf{q}})\sigma_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})$ formulates the null-space projection matrix. The null-space projection of the vector $\dot{\boldsymbol{\xi}}$ does not affect the end-effector position so it can be arbitrary. The

simplest calculation of the pseudo-inverse is:

$$\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\dagger}(\bar{\mathbf{q}}) = \boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}}) \left[\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}}) \boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}}) \right]^{-1}.$$
(6.56)

One possibility of resolving kinematic redundancy is to simply use the pseudo-inverse of the Jacobian for the resolution of the inverse kinematical problem and the vector $\dot{\boldsymbol{\xi}}$ is chosen to be zero.

The use of the simple pseudo inverse (6.56) does not have phisical meaning and may lead to badly conditioned numerical calculations. A mechanically correct generalized inverse of the Jacobian can be interpreted by employing the mass matrix [28]:

$$\boldsymbol{\sigma}_{\bar{\mathbf{q}},\mathbf{M}}^{\dagger}(\bar{\mathbf{q}}) = \mathbf{M}^{-1}(\bar{\mathbf{q}})\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}}) \left(\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\mathbf{M}^{-1}(\bar{\mathbf{q}})\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}})\right)^{-1}.$$
(6.57)

It has been proved [26] that the use of (6.57) in the inverse calculation minimizes the kinetic energy.

In a generalized case we can say that the weighted pseudo-inverse minimizes the cost function defined as $\dot{\mathbf{q}}^{\mathrm{T}} \mathbf{W} \dot{\mathbf{q}}$ with the arbitrarily chosen matrix \mathbf{W} [26], and

$$\boldsymbol{\sigma}_{\bar{\mathbf{q}},\mathbf{W}}^{\dagger}(\bar{\mathbf{q}}) = \mathbf{W}^{-1}\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}}) \left(\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\mathbf{W}^{-1}\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}})\right)^{-1}.$$
(6.58)

Optimization with respect to a scalar performance criterion can be achieved based on the nullspace projection of an arbitrarily chosen joint velocity vector $\dot{\boldsymbol{\xi}}$ (see equation (6.55)). We define the position-dependent scalar performance criterion $p(\bar{\mathbf{q}})$. The null-space vector that minimizes $p(\bar{\mathbf{q}})$ is [26]:

$$\dot{\boldsymbol{\xi}} = k \frac{\partial p(\bar{\mathbf{q}})}{\partial \bar{\mathbf{q}}},\tag{6.59}$$

where k is an arbitrary chosen constant and (6.59) is substituted into (6.55).

The scalar performance criterion $p(\bar{\mathbf{q}})$ can be used for several goals. Let us consider the example, when the goal is to avoid the joint limits. In this case, the criterion has the form:

$$p = \sum_{i=1}^{n} \left[\frac{\varphi_i - \varphi_i^{mid}}{\varphi_i^{mid} - \varphi_i^{max}} \right], \tag{6.60}$$

where the optimal joint angle vector is interpreted as $\varphi_i^{mid} = 0.5(\varphi_i^{min} + \varphi_i^{max})$. Then the calculated motion goes as near to the optimal joint angle as the system allows.

As an other example, the maximization of the *manipulability* is mentioned. Manipulability measures the ability of the manipulator to move uniformly in all directions. At the singular points, the manipulator loses one or more degrees of freedom and the manipulability goes to zero. In other words some tasks cannot be performed at or near singular points. A quantitative measure of the manipulability of a robot is defined as [16]:

$$\omega(\bar{\mathbf{q}}) = \sqrt{\det\left(\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\mathrm{T}}(\bar{\mathbf{q}})\right)}.$$
(6.61)

A goal can be to maximize the manipulability $\omega(\bar{\mathbf{q}})$, which means the manipulator tries to avoid the singular configurations, hence the algorithm minimizes $-\omega(\bar{\mathbf{q}})$. Thus the scalar performance criterion is $p(\bar{\mathbf{q}}) = -\omega(\bar{\mathbf{q}})$.

Sometimes, the *dynamic manipulability* is used as performance criterion $p(\bar{\mathbf{q}}) = -\omega_{d}(\bar{\mathbf{q}})$, which is defined as:

$$\omega_{\rm d}(\bar{\mathbf{q}}) = \sqrt{\det\left(\boldsymbol{\sigma}_{\bar{\mathbf{q}}}(\bar{\mathbf{q}})\tilde{\mathbf{M}}^{\rm T}(\bar{\mathbf{q}})\tilde{\mathbf{M}}(\bar{\mathbf{q}})\boldsymbol{\sigma}_{\bar{\mathbf{q}}}^{\rm T}(\bar{\mathbf{q}})\right)},\tag{6.62}$$

where $\mathbf{M}(\mathbf{\bar{q}})$ is the mass matrix weighted according to the torque limits. The minimization of required torques also can be a criterion [26, 84].

Another common method is when the potential energy $p(\bar{\mathbf{q}}) = U_{v}(\bar{\mathbf{q}})$ of virtual springs is minimized [26, 85]. This idea will be used in Section 6.6.4.

An alternative way that is also used in the literature [26], when the nonlinear algebraic equations defined by (6.52) are augmented directly by the necessary number of algebraic equations which are responsible for minimizing the potential function of the virtual springs:

$$\frac{\partial U_{\mathbf{v}}(\bar{\mathbf{q}})}{\partial \bar{q}_i} = 0.$$
(6.63)

The resulting system is solved by any method applicable for finding roots of nonlinear functions, such as Newton-Raphson method.

6.6.2 Notion of redundancy in case of underactuated robots

Since the notions of both redundancy and underactuation appear in this study, we try to give a classification of the different combinations of these ideas. In order to do this, the definition of dynamical redundancy is proposed [86].

An underactuated manipulator equipped with more independent control inputs than required to perform a specified task is called dynamically redundant underactuated system.

This definition is equivalent to the kinematic redundancy for fully actuated robots. In contrast, the inverse kinematics of underactuated systems cannot be solved uniquely, so these are always kinematically redundant. However, if the task dimension l is equal to the number g of independent actuators, the determination of the control input is unique, and consequently the kinematics can also be calculated uniquely. These systems are dynamically not redundant, the inverse dynamics can be solved uniquely. If the task dimension l is less than the number g of actuators, the system satisfies the above definition of dynamic reundancy, since even the inverse dynamic calculation is not unique, including the control input \mathbf{u} . Table 6.1. summarizes the possible cases for n - m DoF, where n is the number of descriptor coordinates and m is the number of geometric constrainsts.

6.6.3 Computed torque method for dynamically redundant systems

The dynamical model can be written in the form of a differential-algebraic equation defined by equations (3.1) and (3.2). The equations of motion (3.1) and (3.2) are complemented by the servo-constraint equation (3.3) and since the system is dynamically redundant, that is, g > l, an additional optimization rule $\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \in \mathbb{R}^{g-l}$ can be set in the following form [86]:

$$\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}. \tag{6.64}$$

	underactuated	kinematically redundant	dynamically redundant
n-m = g = l	no	no	no
n-m > g = l	yes	yes	no
n-m>g>l	yes	yes	yes

TABLE 6.1. Interpretations of redundancy (n - number of descriptor coordinates, m - the number of geometric constraints, l - dimension of task, g - number of independent actuators)

The optimization rule is given in a similar form as the geometric constraint (3.2) and the servoconstraint (3.3), but it may depend on time explicitly and we assume that the generalized velocity $\dot{\mathbf{q}}$ appears in it, thus the optimization rule is a *rheonomic* and *non-holonomic* constraint.

We assume that the geometric constraints (3.2), the servo-constraints (3.2) and the optimization rule (6.64) are linearly independent and consistent, furthermore (3.3) and (6.64) can be satisfied with bounded control input.

The optimization rule (6.64) is formulated by a non-holonomic constraint, hence the acceleration $\ddot{\mathbf{q}}$ appears in its first time derivative:

$$\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \boldsymbol{\gamma}_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} + \boldsymbol{\gamma}_{t}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}, \tag{6.65}$$

where $\gamma_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t) \in \mathbb{R}^{(g-l) \times n}$ and $\gamma_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}, t) \in \mathbb{R}^{(g-l) \times n}$ are the Jacobians of $\gamma(\mathbf{q}, \dot{\mathbf{q}}, t)$ regarding \mathbf{q} and $\dot{\mathbf{q}}$ respectively. Vector $\gamma_t(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the partial time derivative of the explicitly time dependent part of (6.64). Since the optimization rule is given in the form of an artificial constraint, (6.65) has to be stabilized similarly to the servo-constraint. We extend (6.65) with the positive definite gain matrix \mathbf{K}_{γ} :

$$\gamma_{\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \gamma_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} + \gamma_t(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{K}_{\gamma}\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}.$$
(6.66)

The unconstrained dynamic equation (3.1), the acceleration level geometric constraint equation (3.26), the stabilized, acceleration level servo-constraint equation (3.31) and the stabilized, acceleration level optimization rule (6.66) is incorporated in a hyper-matrix form as follows [86]:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \boldsymbol{\varphi}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t) & -\mathbf{H}(\mathbf{q}) \\ \boldsymbol{\varphi}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t) & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\gamma}_{\dot{\mathbf{q}}}(\mathbf{q},\dot{\mathbf{q}},t) & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \\ \mathbf{u} \end{bmatrix} = \\ \begin{bmatrix} \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \\ -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) \\ -\dot{\boldsymbol{\varphi}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\varphi}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) \\ -\dot{\boldsymbol{\sigma}}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \dot{\boldsymbol{\sigma}}_{t}(\mathbf{q},\dot{\mathbf{q}},t) - \mathbf{K}_{\alpha}[\boldsymbol{\sigma}_{\mathbf{q}}(\mathbf{q},t)\dot{\mathbf{q}} + \boldsymbol{\sigma}_{t}(\mathbf{q},t)] - \mathbf{K}_{\beta}\boldsymbol{\sigma}(\mathbf{q},t) \\ -\boldsymbol{\gamma}_{\mathbf{q}}(\mathbf{q},\dot{\mathbf{q}},t)\dot{\mathbf{q}} - \boldsymbol{\gamma}_{t}(\mathbf{q},\dot{\mathbf{q}},t) - \mathbf{K}_{\gamma}\boldsymbol{\gamma}(\mathbf{q},\dot{\mathbf{q}},t) \end{bmatrix} .$$
(6.67)

In conclusion, the dynamic equation (3.1) is augmented with three different types of constraint equations:

- · geometric constraint equations: holonomic constraints, no stabilization;
- · servo-constraint equations: holonomic constraints, stabilization with gains $\mathbf{K}_{\alpha}, \mathbf{K}_{\beta}$;
- · velocity level servo-constraint equations: non-holonomic constraints, stabilization with \mathbf{K}_{γ} .

6.6.4 Case study for the planar Acroboter

In order to get an overview of the developed control strategy for dynamically redundant multibody systems, consider the simplified planar model of the Acroboter manipulator shown in Fig. 6.14a. The CU is substituted by a single horizontal linear drive, the CC is modeled by a particle with 2 DoF and the SU is modeled by a rigid body with 3 DoF. We handle the manipulator as a multibody system described by the Cartesian coordinates of the base points P_L , P_{cc} , P_1 and P_2 of the bodies involved. Such set of non-minimum set of descriptor coordinates is also called natural coordinates in [36]. The vector of the n = 7 descriptor coordinates and the m = 1 dimensional single geometric constraint representing the constant distance between P_1 and P_2 are introduced as:

$$\mathbf{q} = \begin{bmatrix} x_{\mathrm{L}} & x_{\mathrm{cc}} & z_{\mathrm{cc}} & x_1 & z_1 & x_2 & z_2 \end{bmatrix}^{\mathrm{T}}, \qquad (6.68)$$

$$\varphi(\mathbf{q}) = \left[(x_2 - x_1)^2 + (z_2 - z_1)^2 - L_{34}^2 \right].$$
(6.69)

The system is controlled by g = 5 actuators, which is less than the n - m = 6 DoF, thus the system is underactuated. The actuator forces are shown in Fig. 6.14b and are arranged in the g = 5 dimensional control input vector **u**:

$$\mathbf{u} = \begin{bmatrix} F_{\mathrm{L}} & F_{\mathrm{M}} & F_{1} & F_{2} & F_{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}.$$
(6.70)

The task of the manipulator is to move a specified point of the SU half way between P_1 and P_2 on a prescribed trajectory given by x^d and z^d as functions of time. Besides, the cable connector has to be kept in a given vertical distance h_{cc}^d above the same specified point of the SU, and it is also desired that the SU has to be kept horizontal. These tasks are expressed by the l = 4 dimensional servo-constraint vector:

$$\boldsymbol{\sigma}(\mathbf{q},t) = \begin{bmatrix} \frac{z_1 + z_2}{2} + h_{\rm cc}^{\rm d} - z_{\rm cc} \\ \frac{x_1 + x_2}{2} - x^{\rm d} \\ \frac{z_1 + z_2}{2} - z^{\rm d} \\ z_1 - z_2 \end{bmatrix}.$$
(6.71)

In the given case l < g < n - m, thus the system is underactuated and kinematically and dynamically redundant. One can observe that in (6.71), there are no prescriptions for the $x_{\rm L}$ position of the linear drive. For the redundancy resolution, we use the idea of virtual springs adopted from [85]. The angle of the main cable is minimized, thus we apply a virtual spring between the horizontal positions $x_{\rm L}$ and $x_{\rm cc}$ with virtual stiffness $k_{\rm v}$. Additionally, the speed of the linear drive should be small, so we introduce a virtual damping with damping factor $d_{\rm v}$. The resulting optimization rule can



FIGURE 6.14. Planar model (a) and free body diagrams (b) of the Acroboter

be formulated by the g - l = 1 dimensional non-holonomic servo-constraint equation $\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$ with

$$\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) = \left[d_{\mathrm{v}} \dot{x}_{\mathrm{L}} + k_{\mathrm{v}} (x_{\mathrm{L}} - x_{\mathrm{cc}}) \right].$$
(6.72)

A round-cornered rectangular shaped trajectory was prescribed by means of x^{d} and z^{d} , on which the SU was moved along periodically, while it was prescribed to stay horizontal with a constant CC height h_{cc} . Fig. 6.15 shows one period of the motion. In Fig. 6.16 the time histories of the control forces, the 4 servo-constraint violations and the time history of the violation of the optimization rule can be seen in approximately 2.5 periods. It can be observed that the perturbation applied in the initial time instant is eliminated quickly by the controller, however $\sigma_1...\sigma_4$ and γ_1 values temporarily increase the corners of the prescribed trajectory, where large accelerations are required.

As a summary of the simulation work, we can conclude that the violation of the servo-constraints and the optimization rule could be driven to zero in a stable way, and the motion of the manipulator is also stable and smooth. The numerical simulations indicate, that the method is efficient and feasible for real time applications.



FIGURE 6.15. Simulation results: motion of the Acroboter platform



FIGURE 6.16. Simulation results: time histories

6.7 New results

Thesis 6.

Analytical solution was derived for the inverse dynamics of the Acroboter platform in planar case when the desired orientation of the swinging unit is horizontal during operation. The corresponding CTC algorithms were also tested experimentally on a swinging unit attached to an industrial robot. The inverse dynamical calculation was accomplished also for the spatial multiple mathematical pendulum in closed form, when the trajectory of the lower endpoint is prescribed.

It was shown that the closed form formulae for the desired motion of the actuator and for the desired control forces are identical for the planar Acroboter model and the double pendulum. The results were generalized for the spatial Acroboter model and the simulations indicated that the results coincide with those of the spatial double pendulum. This means that a simplified double pendulum model can be used in the computed torque control of the Acroboter platform. This way, less computational time is needed and the efficiency of the real-time control is improved.

The analytical results can also be used for the validation of numerical results when alternative control methods are tested, like in case of the benchmark problem of underactuated multiple pendulum.

Related publications: [14, 73, 78, 81-83]

Thesis 7.

In case of underactuated systems, it was shown that the classical definition of kinematic redundancy, which is based on the comparison of the degrees of freedom and the dimension of the prescribed task, is not equivalent to the non-uniqueness of the inverse dynamic problem, while it guarantees the non-uniqueness of the inverse kinematic problem. To clarify this issue, the notion of dynamic redundancy was introduced, which is related to the comparison of the dimension of the prescribed task and the number of the actuators: if the number of degrees of freedom is larger than the number of actuators, and the latter is larger than the dimension of the prescribed task, then the inverse dynamic problem has no unique solution either. Accordingly, the computed torque control method was generalized to dynamically redundant systems: the geometric- and servo-constraints were augmented by the so-called velocity level (non-holonomic) servo-constraints in a way that the inverse dynamic problem was solvable uniquely and the total mechanical energy was also minimized similarly to the pseudo-inverse calculations.

Related publication: [86]

Bibliography

- A. Zelei, L. Bencsik, L. L. Kovács, and G. Stépán, "Impact models for walking and running systemsangular moment conservation versus varying geomteric constraints," in *In proc.: Eccomas 2013*, July 1-4, 2013, Zagreb, Croatia, pp. 47–48.
- [2] A. Zelei, L. Bencsik, L. L. Kovács, and G. Stépán, "Energy efficient walking and running impact dynamics based on varying geometric constraints," in *In proc.: 12th Conference on Dynamical Systems Theory and Applications - DSTA 2013*, December 2-5, 2013, Lodz, Poland, pp. 333–334.
- [3] R. Olfati-Saber, Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles, Ph.D. thesis, Massachusetts Institute of Technology, 2001.
- [4] R. J. Henry, Z. N. Masoud, A. H. Nayfeh, and D. T. Mook, "Cargo pendulation reduction on ship-mounted cranes via boom-lu angle actuation," *Journal of Vibration and Control*, vol. 7, no. 8, pp. 1253–1264, 2001.
- [5] H. M. Omar, *Control of gantry and tower cranes*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 2003, supervisor: A. H. Nayfeh.
- [6] W. Blajer and K. Kolodziejczyk, "Modeling of underactuated mechanical systems in partly specified motion," *Journal of Theoretical and Applied Mechanics*, vol. 46, no. 2, pp. 383–394, 2008.
- [7] I. M. M. Lammerts, Adaptive Computed Reference Computed Torque Control, Ph.D. thesis, Eindhoven University of Technology, 1993.
- [8] H. Goldstein, Classical Mechanics, 2nd ed. Addison-Wesley, 1980.
- G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators," in 30th Conference on Decision and Control, Brighton, England, 1991, pp. 2398–2403.
- [10] W. S. Levine (Ed.), Control Handbook, CRC and IEEE, 1996.
- [11] R. Seifried, Dynamics of Underactuated Multibody Systems: Modeling, Control and Optimal Design, Springer, 2014.
- [12] G. Stépán and et al., "A ceiling based crawling, hoisting and swinging service robot platform," in Proceedings of Beyond Gray Droids: Domestic Robot Design for the 21st Century Workshop at HCI 2009, 2009.
- [13] W. Blajer and K. Kolodziejczyk, "Control of underactuated mechanical systems with servo-constraints," *Nonlinear Dynamics*, vol. 50, no. 4, pp. 781–791, 2007, From the issue entitled "Dynamical Systems: Theory and Applications".
- [14] A. Zelei, L. L. Kovács, and G. Stépán, "Computed torque control of an under-actuated service robot platform modeled by natural coordinates," *Commun Nonlinear Sci Numer Simulat*, vol. 16, no. 5, pp. 2205–2217, 2011.

- [15] B. Siciliano and O. Khatib (eds.), Handbook of Robotics, Springer, 2008, ISBN 978-3-540-23957-4.
- [16] M. W. Spong and M. Vidyasagar, Robot Dynamics and Control, John Wiley & Sons, 1989.
- [17] R. M. Murray, Z. Li, and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994, ISBN 978-0-849-37981-9.
- [18] da Vinci Surgery, "da Vinci Surgical System," http://www.davincisurgery.com/, Last accessed: July 2011.
- [19] A. Isidori, Nonlinear Control Systems, 3rd ed. Springer Verlag, 1995.
- [20] W. Blajer, K. Dziewiecki, K. Kolodziejczyk, and Z. Mazur, "Inverse dynamics of a two-degree freedom underactuated system: Analysis and experiment," in 10th Conference on Dynam Systems - Theory and Applications DSTA-2009, December 7-10 Lodz, Poland, 2009.
- [21] P. G. H. Tackenberg, "Control methods for vibration reduction in an underactuated nonlinear beam system," M.S. thesis, Eindhoven University of Technology, 1997.
- [22] A. De Luca, R. Mattone, and G. Oriolo, "Control of redundant robots under end-effector commands: A case study in underactuated systems," *Applied Mathematics and Computer Science*, vol. 7, no. 2, pp. 225–251, 1997.
- [23] N. P. I. Aneke, Control of underactuated mechanical systems, Ph.D. thesis, Technische Universiteit Eindhoven, 2002.
- [24] A. Bommer, "Control of underactuated mechanical systems," Tech. Rep., Technische Universiteit Eindhoven, 2004.
- [25] M.W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *IROS'94*, Munich, Germany, 1994, pp. 314–321.
- [26] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," in Int. Conf. Robotics and Automation, St. Louis, MS, 1985, pp. 1016–1021.
- [27] Y. Nakamura, Advanced Robotics: Redundancy and Optimization, 1st. ed, Addison-Wesley Longman Publishing Co., 1990.
- [28] J. Kövecses, J.-C. Piedoboeuf, and C. Lange, "Dynamic modeling and simulation of constrained robotic systems," *IEEE/ASME Transactions on mechatronics*, vol. 8, no. 2, pp. 165–177, 2003.
- [29] K.M. Hangos, J. Bokor, and G. Szederkényi, Analysis and Control of Nonlinear Process Systems, Springer-Verlag, London, 2004.
- [30] V. D. Yurkevich, Design of Nonlinear Control Systems with the Highest Derivative in Feedback, World Scientific Publishing Co., Singapore, 2004.
- [31] W. Blajer and K. Kolodziejczyk, "A geometric approach to solving problems of control constraints: Theory and a dae framework," *Multibody System Dynamics*, vol. 11, no. 4, pp. 343–364, 2004.
- [32] A. Zelei and G. Stépán, "Case studies for computed torque control of constrained underactuated systems," *Periodica Polytechnica - Mechanical Engineering*, vol. 56, no. 1, pp. 73–80, 2012.
- [33] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer-Verlag, Berlin (2nd ed.), 1996.

- [34] U. M. Ascher and L. R. Petzold, Computer Methods for Ordinary Differential equations and Differential-Algebraic equations, Philadelphia: SIAM. ISBN 978-0-89871-412-8., 1998.
- [35] L. L. Kovács, A. Zelei, L. Bencsik, J. Turi, and G. Stépán, "Motion control of an under-actuated service robot using natural coordinates," in *Proceedings of ROMANSY 18 - Robot Design, Dynamics and Control*, July 5 - 8, 2010, Udine, Italy, pp. 331–338.
- [36] J.G. de Jalón and E. Bayo, Kinematic and dynamic simulation of multibody systems: the real-time challenge, Springer-Verlag, 1994.
- [37] V. I. Kirgetov, "The motion of controlled mechanical systems with prescribed constraints (servo constraints)," *Prikl. Mat. Mekh.*, vol. 31, no. 3, pp. 433–447, 1967.
- [38] L. L. Kovács and A. Zelei, "Servo constraint based computed torque control of acroboter," in In proc. of the Symposium on Brain, Body and Machine on the occasion of the 25th anniversary of McGill University's Centre for Intelligent Machines, November 10 - 12, 2010, Montreal, Canada, p. 4 pages.
- [39] L. L. Kovács, J. Kövecses A. Zelei, L. Bencsik, and G. Stépán, "Servo-constraint based computed torque control of underactuated mechanical systems," in *In proc. of ASME 2011 8th International Conference* on Multibody Systems, Nonlinear Dynamics, and Control, August 28 - 31, 2011, Washington, USA, pp. 331–338.
- [40] L. Bencsik, A. Zelei, and L. L. Kovács, "Alulaktuált robotok kiszámított nyomaték szabályozása szervo kényszerek alkalmazásával," in *In proc.: XI. Magyar Mechanikai Konferencia*, August 29-31, 2011, Miskolc, Hungary, pp. 1 page (Extended abstract), paper id.: 114.
- [41] L. Bencsik, L. L. Kovács, and A. Zelei, "Periodic servo-constraints for stabilizing underactuated multibody systems," in In proc. of the The 3rd Joint International Conference on Multibody System Dynamics and The 7th Asian Conference on Multibody Dynamics, June 30-July 3, 2014, Busan, Korea, p. 2 pages.
- [42] L. Bencsik and L. L. Kovács, "Stability case study of an underactuated service robot," in Dynamical Systems - Nonlinear Dynamics and Control, DSTA Dec. 5-8, 2011, Lodz, Poland, pp. 89–94.
- [43] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," Computer methods in applied mechanics and engineering, vol. 1, no. 1, pp. 1–16, 1972.
- [44] P. Flores, M. Machado, E. Seabra, and M. T. da Silva, "A parametric study on the baumgarte stabilization method for forward dynamics of constrained multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 6, no. 1, 2010.
- [45] A. Zelei, L. Bencsik, and G. Stépán, "An underactuated modular robot for testing control algorithms," in *In proc.: 56th IWK International Scientific Colloquium*, 12-16 September, 2011, Ilmenau, Germany.
- [46] A. Zelei, L. L. Kovács, L. Bencsik, and G. Stépán, "Computed torque control of under-actuated dynamical systems modeled by natural coordinates," in *In Proceedings of the 7th International Conference on Mechanical Engineering. Gépészet 2010*, 25-26. May, 2010, Budapest, Hungary, pp. 11–18, vol. 30.
- [47] A. Zelei and G. Stépán, "The acroboter platform part 2: Servo-constraints in computed torque control," in In proceedings of the IUTAM Symposium on Dynamics Modeling and Interaction Control in Virtual and Real Environments - IUTAM Book Series by Springer, 7-11 June, 2010, Budapest, Hungary, pp. 11-18, vol. 30.
- [48] A. Zelei, M. Móra, and G. Stépán, "Kísérletek ventilátorral hajtott alulaktuált inga szabályozására," in In proc.: XI. Magyar Mechanikai Konferencia, August 29-31, 2011, Miskolc, Hungary, pp. 6 pages, paper id.: 138.

- [49] A. Zelei, M. Móra, and G. Stépán, "Experimental analysis of a fan driven underactuated pendulum," in In proc. of the 28th Danubia-Adria-Symposium on Advances in Experimental Mechanics - DAS 2011, 28 September - 1 October, 2011, Siófok, Hungary, p. 2 pages.
- [50] E. T. Enikov, Aeropendulum project, University of Arizona, ," http://aeropendulum.arizona.edu/, Last accessed: Jan. 2015.
- [51] Z. Juhasz and A. Zelei, "Analysis of worm-like locomotion," Periodica Polytechnica Mechanical Engineering, vol. 57, no. 2, pp. 59–64, 2013.
- [52] J. Steigenberger, "Some theory towards a stringent definition of locomotion," *Multibody System Dynamics*, vol. 26, no. 1, pp. 81–90, 2011.
- [53] R. L. Lieber and S. C. Bodine-Fowler, "Skeletal muscle mechanics: Implications for rehabilitation," *Physical Therapy*, vol. 73, pp. 844–856, 1993.
- [54] P. Lu, "Tracking control of nonlinear systems with bounded controls and control rates," Automatica, vol. 33, no. 6, pp. 1199–1202, 1997.
- [55] W. E. Dixon, M. S. de Queiroz, F. Zhang, and D. M. Dawson, "Tracking control of robot manipulators with bounded torque inputs," *Robotica*, vol. 17, no. 2, pp. 121–129, 1999.
- [56] W. Peng, Z. Lin, and J. Su, "Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques," *IET Control Theory and Applications*, vol. 3, no. 6, pp. 701–711, 2009.
- [57] A. Zelei and G. Stépán, "Computed torque control of a constrained manipulator considering the actuator saturation," in proc. of 11th Conference on Dynamical Systems - Theory and Applications DSTA-2011 (Dynamical Systems - Nonlinear Dynamics and Control), December 5-8, 2011, Lodz, Poland, pp. 189–194.
- [58] D. Cunningham and H. H. Asada, "The winch-bot: A cable-suspended, under-actuated robot utilizing parametric self-excitation," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17, 2009.
- [59] A. Zelei, L. Bencsik, and G. Stépán, "Alternative task definitions for path tracking control of underactuated robots," in Book of Abstracts of the The 3rd Joint International Conference on Multibody System Dynamics and The 7th Asian Conference on Multibody Dynamics, June 30-July 3, 2014, Busan, Korea, p. 2 pages.
- [60] L. L. Kovács and L. Bencsik, "Stability case study of the acroboter underactuated service robot," Theoretical and Applied Mechanics Letters, vol. 2, no. 043004, pp. 7 pages, 2012, DOI: 10.1063/2.1204304.
- [61] A. Zelei and G. Stépán, "The influence of parametric excitation on floating bodies," in Proceedings of the 79th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM 2008), 31. March - 4. April, 2008, Bremen, Germany.
- [62] A. Zelei and G. Stépán, "Stabilizing effect of vertical periodic motion on athletes paddle systems," in In: Proceedings of the 3rd Hungarian Confrance on Biomechanics, 2008, Budapest, Hungary, pp. 415–422.
- [63] A. Zelei and G. Stépán, "The influence of parametric excitation on floating bodies," PAMM, vol. 8, no. 1, pp. 10929–10930, 2008.
- [64] Á. Takács and A. Zelei, "U'szó testek stabilitásvesztése hullámzás hatására," in In proc: XI. Magyar Mechanikai Konferencia, August 29-31, 2011, Miskolc, Hungary, pp. paper id: 066, extended abstract.

- [65] M. G. Floquet, "équations différentielles linéaires a coefficients péridiques," Ann. Sci. Ec. Normale Super., vol. 12, pp. 47–89, 1883.
- [66] E. Mathieu, "Mémoire sur le mouvement vibratoire d'une membrane de forme elliptique," Ann. Sci. Ec. Normale Super., vol. 13, pp. 137–203, 1868.
- [67] A. Stephenson, "On a new type of dynamical stability," Mem. Proc. Manch. Lit. Phil. Soc., vol. 52, pp. 1–10, 1908.
- [68] T. Insperger and R. Horváth, "Pendulum with harmonic variation of the suspension point," *Periodica Polytechnika Ser. Mech. Eng.*, vol. 44, no. 1, pp. 39–46, 2000.
- [69] F. van der Pol and M. J. O. Strutt, "On the stability of the solutions of mathieu's equation," *Philos. Mag.*, vol. 5, pp. 18–38, 1928.
- [70] M. Farkas, *Periodic Motions*, Springer, New York, 1994.
- [71] M. Begon and F. Colloud, "A kayak ergometer using a sliding trolley to reproduce accurate on-water mechanical conditions," *Journal of Biomechanics*, vol. 40, no. supplement 2, pp. S439, 2007.
- [72] A. Baudouin and D. Hawkins, "A biomechanical review of factors affecting rowing performance," British Journal of Sports Medicine, vol. 36, no. 6, pp. 396–402, 2002.
- [73] A. Zelei and G. Stépán, "Motion analisys of a crane as an under-actuated robot," in Gépészet 2008: Proceedings of Sixth Conference on Mechanical Engineering., 2008, Budapest, Hungary, Paper no.: G-2008-M-06, ISBN: 978-963-420-947-8.
- [74] Z. N. Masoud, A Control System for the Reduction of Cargo Pendulation of Ship-Mounted Cranes, Ph.D. thesis, Virginia Polytechnic Institute and State University, 2000, supervisor: A. H. Nayfeh.
- [75] L. Zhang, H. Wang, and H. Hu, "Symbolic computation of normal form for hopf bifurcation in a neutral delay differential equation and an application to a controlled crane," *Nonlinear Dynamics*, vol. 70, no. 1, pp. 463–473, 2012.
- [76] B. Kimiaghalam, A. Homaifar, and M. Bikdash, "Fuzzy dynamic friction controller for a ship crane," in The 6th International Conference on Fuzzy Theory and Technology, 1998, pp. 1:203–206.
- [77] K. Nakazono, K. Ohnisihi, and H. Kinjo, "Load swing suppression in jib crane systems using a genetic algorithm-trained neuro-controller," in *The 4th IEEE International Conference on Mechatronics, ICM* '07-Kumamoto, May 8-10, 2007, pp. WA1–C–1, 1–4.
- [78] L. L. Kovács, A. Zelei, L. Bencsik, and G. Stépán, "The acroboter platform part 1: Conceptual design and dynamics modeling aspects," in *In proceedings of the IUTAM Symposium on Dynamics Modeling* and Interaction Control in Virtual and Real Environments - IUTAM Book Series by Springer, 7-11 June, 2010, Budapest, Hungary, pp. 3–10, vol. 30.
- [79] T. Sato, R. Fukui, H. Mofushita, and T. Mori, "Construction of ceiling adsorbed mobile robots platform utilizing permanent magnet inductive traction method," in *Proceedings of 2004 IEEEiRSJ International Conference on Intelligent Robots and Systems, September 28 - October 2, Sendai, Japan, 2004, pp. 552– 558.*
- [80] P. J. McKerrow and D. Ratner, "The design of a tethered aerial robot," in Proceedings of IEEE International Conference on Robotics and Automation, Rome, Italy, 10-14 April, 2007, pp. 355–360.
- [81] A. Zelei, L. Bencsik, G. Stépán, and L. L. Kovács, "Dynamics and actuation of the acroboter platform," in Book of Abstracts of The 2nd Joint International Conference on Multibody System Dynamics - IMSD 2012, May 29 - June 1, 2012, Stuttgart, Germany, pp. 333–334.

- [82] A. Zelei, L. L. Kovács, and G. Stépán, "Trajectory generation of an underactuated and redundant service robot platform equipped with ducted fan actuators," in *In: Proceedings of the 10th Conference on Dynamical Systems - Theory and Applications DSTA-2009*, December 7 - 10, 2009, Lodz, Poland, pp. 789–796.
- [83] A. Zelei and G. Stépán, "Computed torque control method for under-actuated manipulator," in Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, California USA, 2009, pp. Paper id: DETC2009– 86409.
- [84] A. S. Deo and I. D. Walker, "Minimum effort inverse kinematics for redundant manipulators," IEEE Trans. Rob. Automation, vol. 13, no. 5, pp. 767–775, 1997.
- [85] A. McLean and S. Cameron., "The virtual springs method: Path planning and collision avoidance for redundant manipulators," *Intl. Journal of Robotics Research*, vol. 15, no. 4, pp. 300–319, 1996.
- [86] A. Zelei, L. Bencsik, L. L. Kovács, and G. Stépán, "Redundancy resolution of the underactuated manipulator acroboter," in *In proc. RoManSy 2012 - 19th CISM-IFToMM Symposium on Robot Design*, *Dynamics and Control*, June 12-15, 2012, Paris, France, pp. 233–240.