

Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Planning and Control Methods for Robotic Manipulation Tasks with Non-Negligible Dynamics

Georg Rudolf Sebastian Bätz

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Klaus Diepold

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Univ.-Prof. Dr.-Ing., Dr.-Ing. habil. Alois Knoll

Die Dissertation wurde am 26.04.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 15.12.2011 angenommen.

Foreword

The last three and a half years at the Institute of Automatic Control Engineering (LSR) at TU München have been a very exciting time. This dissertation summarizes a good part of the research work I conducted during this time. This thesis work would not have been possible without numerous people that have helped and supported me. First and foremost, I would like to thank my advisor Prof. Martin Buss, for stimulating discussions and guidance throughout my time as a PhD student. My sincere thanks also go to my co-advisor, Dr. Dirk Wollherr, for his support and encouragement during the last three years. Also, I would like to thank Dr. Kolja Kühnlenz for his helpful advice on numerous problems related to computer vision and image processing. The control design for the dribbling task with a compliant end effector in the fourth chapter of this thesis is the result of a collaboration with Prof. Anton Shiriaev and Dr. Uwe Mettin. I would like to thank both for the insightful discussions and the productive cooperation. Furthermore, I am greatly indebted to the German National Academic Foundation which supported me throughout my graduate studies.

The research stay in the LIMS lab at Northwestern University in Evanston at the end of my PhD was an extremely productive and stimulating experience. I would like to thank Prof. Kevin Lynch for providing this opportunity and for the fruitful discussions during my stay.

Daily life at the LSR was a very pleasant experience and the great colleagues were a prominent reason for this. First of all, this refers to my two office mates Raphaela Groten and Klaas Klasing. Thank you, Raphi and Klaas, not only for all the helpful and motivating discussions but also for the lots of laughter and funny moments that we shared. Also, I would like to thank Michael Scheint, Thomas Schauß, Ulrich Unterhinninghofen, Markus Rank, Matthias Rungger, Haiyan Wu, and Kwang-Kyu Lee for the stimulating scientific discussions, the helpfulness, and the enjoyable lunch breaks. Several students contributed to this work: I thank Lorenz Kniep, Arhan Yaqub, Alexander Schmidts, Xihua Lu, Andreas Achhammer, and Nicolas Lehment for their efforts.

Finally, the biggest thank goes to my family for supporting me without reservation.

Munich, April 2011

Georg Bätz

Contents

1	Introduction	1
1.1	Overview of Manipulation	2
1.2	Applications	6
1.3	Control Framework and General Approach	7
1.4	Contributions and Outline of this Thesis	9
2	Modeling Foundations for Dynamic Object Manipulation	11
2.1	Preliminaries	12
2.2	Hybrid System Model	12
2.3	Kinematic Constraints	14
2.4	Contact Kinematics	15
2.4.1	Sliding	16
2.4.2	Rolling	16
2.4.3	Spinning	17
2.4.4	No Contact / Free-Flight	17
2.5	Friction	17
2.5.1	Static Models	17
2.5.2	Dynamic Models	19
2.6	Impacts	20
2.6.1	Classification Criteria	20
2.6.2	Discrete Models	22
2.6.3	Continuous Models	24
2.6.4	Extensions	24
2.7	Summary	24
3	Non-Periodic Dynamic Manipulation Tasks	27
3.1	Related Work	28
3.2	Trajectory Planning	29
3.2.1	Constraints	29
3.2.2	Optimization Criteria	30
3.2.3	Optimization Method	31
3.3	Application for Rigid End Effectors	32
3.3.1	Balancing	32
3.3.2	Throwing	34
3.3.3	Nonprehensile Catching	34
3.4	Summary	36

4	Periodic Manipulation Tasks with Intermittent Contact	37
4.1	Related Work	38
4.2	Trajectory Planning	39
4.3	Stability of Periodic Solutions of Ordinary Differential Equations	40
4.4	Stability of Periodic Solutions of Hybrid Dynamical Systems	42
4.4.1	Local Stability Analysis	42
4.4.2	Non-Local Stability Analysis	43
4.5	Application for Rigid End Effectors	44
4.5.1	Classic Juggling Task	44
4.5.2	Dribbling Task	49
4.5.3	Comparison of Classic Juggling and Dribbling	51
4.6	Application for Compliant End Effectors	52
4.7	Summary	59
5	Perception and Interaction Control for Dynamic Manipulation	61
5.1	Related Work	62
5.2	Dynamic Contact Force/Torque Observer	64
5.2.1	Continuous-Time System Model	66
5.2.2	Discrete-Time System Model	67
5.2.3	Filter Design	68
5.2.4	Simulation	70
5.2.5	Experiments	77
5.3	Motion and Interaction Control	82
5.3.1	Task Space Motion Control	82
5.3.2	Interaction Control	83
5.4	High-Speed Vision	85
5.4.1	Object Tracking Module	85
5.4.2	Trajectory Prediction Module	86
5.5	Summary	87
6	Experimental Evaluation	89
6.1	Experimental Setup	90
6.2	Throwing	90
6.3	Nonprehensile Catching	91
6.4	Dribbling	93
6.4.1	Rigid End Effector	94
6.4.2	Compliant End Effector	96
6.4.3	Comparison	97
6.5	Juggling	99
6.6	Summary	99
7	Conclusions	101
7.1	Summary	101
7.2	Discussion and Future Directions	103

A Appendix	105
A.1 Differential Geometry	105
A.2 Velocity and Force/Torque Transformations	106
A.3 Orientation of a Rigid Body: Unit Quaternion	107
A.4 Newton-Euler Equations for a Rigid Body	108
Bibliography	110

Notations

Abbreviations

COM	Center of mass
COR	Coefficient of restitution
DOF	Degrees of freedom
ODE	Ordinary differential equation
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
F/T	Force/Torque
HSM	Hybrid system model
TOE	Topological orbital equivalence

Conventions

Scalars, Vectors, and Matrices

Scalars are denoted by upper and lower case letters in italic type. Scalar forces and torques are denoted by upper case letters in italic type. *Vectors* are denoted by lower case letters in bold type, and the vector \mathbf{x} is composed of elements x_i . Vectorial forces and torques as well as linear and angular momentum are denoted by upper case letters in bold type. *Matrices* are denoted by upper case letters in bold type, and the matrix \mathbf{M} is composed of elements M_{ij} (i^{th} row, j^{th} column).

x	Scalar
X	Scalar force/torque
\mathbf{x}	Vector
\mathbf{X}	Matrix, vectorial force/torque, or linear/angular momentum
$f(\cdot)$	Scalar function
$\mathbf{f}(\cdot)$	Vector function
$\dot{\mathbf{x}}, \ddot{\mathbf{x}}$	time derivatives w.r.t. an inertial frame $\frac{d}{dt}\mathbf{x}$ and $\frac{d^2}{dt^2}\mathbf{x}$
\mathbf{M}^T	Transpose of \mathbf{M}
\mathbf{M}^{-1}	Inverse of \mathbf{M}
\mathbf{M}^\dagger	Pseudoinverse of \mathbf{M}

Symbols

General

\mathbb{R}	Real numbers
\mathbb{Z}	Integers
Σ_0	Inertial frame
Σ_C	Frame located at point C
m_C	Mass of body C
\mathbf{J}_A^C	Inertia tensor w.r.t. point A, expressed in frame Σ_C
T_s	Sampling time, $T_s = 0.001$ s
g	Gravitational acceleration, $g = +9.81\text{m/s}^2 > 0$
\mathbf{g}	Gravitational acceleration vector, $\mathbf{g} = [0 \ 0 \ -g]^T$
c_r	(Newton's) coefficient of restitution
\mathbf{q}	Generalized coordinates or actual joint angles, $\in \mathbb{R}^n$
$\mathbf{S}(\cdot) = \mathbf{S}$	Skew-symmetric operator
\mathbf{R}_1^0	Orientation matrix of Σ_1 w.r.t. Σ_0 ; $\mathbf{v}^0 = \mathbf{R}_1^0 \mathbf{v}^1$
$\mathbf{o}_{10} = \{\eta_{10}, \boldsymbol{\epsilon}_{10}\}$	Unit quaternion, representing the orientation of Σ_1 w.r.t. Σ_0
$\eta_{10}, \boldsymbol{\epsilon}_{10}$	Scalar (vector) part of the unit quaternion
*	Quaternion product operation
$\hat{\mathbf{F}}$	Upper case letters: estimate of quantity \mathbf{F}
$\hat{\mathbf{v}}$	Lower case letters: normalized vector \mathbf{v}
$\ \mathbf{v}\ $	Euclidean norm of \mathbf{v}
T	Periodic time
Σ	Transversal cross section
Γ	Periodic solution (periodic orbit, limit cycle)
$\mathbf{P}(\cdot)$	Poincaré map for an autonomous system
$\phi_t(\cdot)$	Flow of a dynamical system
t^-, t^+	Limit from the left and limit from the right of time t
$\mathbf{x}^-, \mathbf{x}^+$	Vector \mathbf{x} at time t^- and t^+
$\mathbf{x}_{[k]} = \mathbf{x}(t_k)$	Value of \mathbf{x} at time t_k
\mathbf{x}^*	Fixed point

Control Structure

$\mathbf{x}_{M,d}, \mathbf{x}_{M,a}$	Desired / actual manipulator pose, $\in \mathbb{R}^7$
$\dot{\mathbf{x}}_{M,d}, \dot{\mathbf{x}}_{M,a}$	Desired / actual manipulator velocity, $\in \mathbb{R}^6$
$\ddot{\mathbf{x}}_{M,d}, \ddot{\mathbf{x}}_{M,a}$	Desired / actual manipulator acceleration, $\in \mathbb{R}^6$
$\mathbf{x}_{O,(t+\Delta t)}$	Predicted object pose at time $t + \Delta t$, $\in \mathbb{R}^7$
$\dot{\mathbf{x}}_{O,(t+\Delta t)}$	Predicted object velocity at time $t + \Delta t$, $\in \mathbb{R}^6$
$\mathbf{F}_S, \mathbf{M}_S$	Measured sensor forces / torques, $\in \mathbb{R}^3$
$\hat{\mathbf{F}}_E, \hat{\mathbf{M}}_E$	Estimated environment forces / torques, $\in \mathbb{R}^3$
$\mathbf{F}_{E,d}, \mathbf{M}_{E,d}$	Desired environment forces / torques, $\in \mathbb{R}^3$
$\boldsymbol{\tau}$	Motor torques, $\in \mathbb{R}^n$

Hybrid Modeling

\mathbf{x}	Continuous state (vector) of a system
x_d	Discrete state of a system
n_d	Number of discrete states
$\zeta = [\mathbf{x}^T x_d]^T$	Hybrid state vector of a system
\mathbf{y}	Continuous measurement (vector) of a system
y_d	Discrete measurement of a system
\mathbf{u}	Continuous input (vector) of a system
u_d	Discrete input of a system
$s_i^j(\cdot) = 0$	Algebraic description of the transition surface
$\Delta_i^j(\mathbf{x})$	Jump map for a transition from $x_d = i$ to $x_d = j$

Contact Kinematics

M	Metric tensor
\mathbf{n}	Outward pointing unit normal vector
K	Curvature tensor
T	Torsion form
u, v	Local coordinates
Σ_{c1}, Σ_{c2}	Contact frame for body 1 (body 2)
Σ_{li}	Body-fixed frame aligned with contact frame Σ_{ci} at time t
$\mathbf{v} = [v_x \ v_y \ v_z]^T$	Translational velocity of $\Sigma_{l1}(t)$ relative to $\Sigma_{l2}(t)$
$\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$	Rotational velocity of $\Sigma_{l1}(t)$ relative to $\Sigma_{l2}(t)$

Friction

v	Relative velocity of the contact areas
F_n	Normal force
F_e	External force
μ_c	Coulomb friction coefficient
μ_s	Static friction coefficient
k_v	Viscous friction coefficient
$F_c(v, F_n)$	Coulomb friction
$F_v(v)$	Viscous friction
$F_s(F_n, F_e)$	Static friction
$F_{s,t}$	Static friction threshold, break-away force
v_s	Stribeck velocity

Impacts

$\Delta P_n, \Delta P_t$	Linear impulse in normal and tangential direction
$\Delta P_{n,c}, \Delta P_{n,r}$	Linear normal impulse of compression / restitution phase
W_c, W_r	Work in compression / restitution phase

$v_{C,n}, v_{C,t}$	Rel. translational velocity at C in normal / tangential direction
ω_C	Rel. rotational velocity at C
$c_{r,N} = c_r, c_{r,P}, c_{r,S}$	Newton's / Poisson's / Stronge's COR in normal direction
$c_{r,t}$	COR in tangential direction
$c_{r,r}$	COR for rotational velocity

Action Planning and Applications

$c_e(\cdot), c_i(\cdot)$	Equality / inequality constraints
$w_j(\cdot), w_d(\cdot), w_u(\cdot)$	Selection criteria
$J(\cdot)$	Cost function
γ	Parameter set subject to optimization
$x_M, \mathbf{x}_M = [\mathbf{p}_M^T \ \mathbf{o}_M^T]^T$	Pose of the manipulator, $\in \mathbb{R} / \in \mathbb{R}^7$
$\dot{x}_M, \dot{\mathbf{x}}_M = [\mathbf{v}_M^T \ \boldsymbol{\omega}_M^T]^T$	Velocity of the manipulator, $\in \mathbb{R} / \in \mathbb{R}^6$
$\ddot{x}_M, \ddot{\mathbf{x}}_M = [\mathbf{a}_M^T \ \boldsymbol{\alpha}_M^T]^T$	Acceleration of the manipulator, $\in \mathbb{R} / \in \mathbb{R}^6$
$x_B, \mathbf{x}_B = [\mathbf{p}_B^T \ \mathbf{o}_B^T]^T$	Pose of the ball, $\in \mathbb{R} / \in \mathbb{R}^7$
$\dot{x}_B, \dot{\mathbf{x}}_B = [\mathbf{v}_B^T \ \boldsymbol{\omega}_B^T]^T$	Velocity of the ball, $\in \mathbb{R} / \in \mathbb{R}^6$
$\ddot{x}_B, \ddot{\mathbf{x}}_B = [\mathbf{a}_B^T \ \boldsymbol{\alpha}_B^T]^T$	Acceleration of the ball, $\in \mathbb{R} / \in \mathbb{R}^6$
$r_B = d_B/2$	Radius of the ball
m_M, m_B	Mass of the manipulator / ball
\tilde{m}_M, \tilde{m}_B	Merged mass of plate and manipulator / plate and ball
c	Spring constant
$l_c, l_{c,0}, l_{c,min}$	Spring length / at equilibrium / at maximum deflection

Dynamic Force/Torque Observer

$\mathbf{F}_S, \mathbf{M}_S$	Measured sensor forces / torques
$\mathbf{F}_E, \mathbf{M}_E$	Environment forces / torques
$\mathbf{F}_I, \mathbf{M}_I$	Inertial forces / torques
$\mathbf{F}_G, \mathbf{M}_G$	Gravitational forces / torques
\mathbf{p}, \mathbf{o}	Position / orientation of the tool's COM
$\mathbf{v}, \boldsymbol{\omega}$	Translational / rotational velocity of the tool's COM
$\mathbf{a}, \boldsymbol{\alpha}$	Translational / rotational acceleration of the tool's COM
$\mathbf{r}_{CS}, \mathbf{r}_{CE}$	Position vector from point C to point S (to point E)
$\mathbf{w}, \boldsymbol{\nu}$	Process / measurement noise
\mathbf{Q}, \mathbf{R}	Process / measurement noise covariance

Motion and Interaction Control

$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$	Frictional torques, $\in \mathbb{R}^n$
$\mathbf{g}(\mathbf{q})$	Gravitational torques, $\in \mathbb{R}^n$
$\mathbf{J}(\mathbf{q})$	Manipulator Jacobian, $\in \mathbb{R}^{6 \times n}$
$\mathbf{B}(\mathbf{q})$	Inertia matrix $\in \mathbb{R}^{n \times n}$
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$	Coriolis and centrifugal forces, $\in \mathbb{R}^n$

Abstract

This thesis investigates novel planning and control methods for robotic manipulation tasks with non-negligible dynamics. The central goal is to equip robots with advanced sensory and motor skills. This represents an important contribution to the development of flexible and versatile robotic systems that allow natural and intuitive interaction with humans.

The thesis follows a model-based approach and discusses the fundamental, state-of-the-art models for the description of the system and its environment. Methods for optimal motion planning and for hybrid control of dynamic manipulation tasks are presented and applied to a number of case studies such as throwing, catching, and juggling. In addition, ball dribbling is introduced as a novel case study for dynamic dexterity. For the task planning process, a simple and hence generic end effector design is considered. The specific challenges with respect to environment perception are addressed that are characteristic for dynamic object manipulation. Based on the fusion of different sensor modalities, a dynamic force/torque observer for the estimation of environment forces and torques is developed. In addition, a method for high-speed image processing is presented that allows to track and predict the state of the manipulated objects. Together with the robot action planning, the two modules are integrated in an extensive control framework. In order to improve the robot's performance in dynamic manipulation tasks, an intrinsically compliant end effector design is developed and evaluated. The effectiveness of the employed methods and of the overall framework is demonstrated in a number of simulations and experiments.

Zusammenfassung

Diese Dissertation untersucht neuartige Methoden zur Planung und Regelung von dynamischen Manipulationsaufgaben in der Robotik. Zentrales Ziel ist die Weiterentwicklung der sensorischen und motorischen Fähigkeiten von Robotern. Hierdurch wird ein wichtiger Beitrag zur Entwicklung von flexiblen und vielseitigen Robotersystemen geleistet, die eine natürliche und intuitive Interaktion mit dem Menschen erlauben.

Die Arbeit folgt einem modellbasierten Ansatz und diskutiert die grundlegenden System- und Umgebungsbeschreibungen. Es werden Methoden zur optimalen Bewegungsplanung sowie zur hybriden Regelung von dynamischen Manipulationsaufgaben präsentiert und auf eine Reihe von Fallbeispielen wie Werfen, Fangen oder Jonglieren angewendet. Zudem wird das Dribbeln eines Balls als neue Fallstudie zur Untersuchung von dynamischer Geschicklichkeit eingeführt. Des Weiteren werden die Probleme im Bereich der Perzeption behandelt, welche speziell bei dynamischer Objektmanipulation auftreten. Basierend auf der Integration verschiedener Sensormodalitäten wird der Entwurf eines Beobachters zur Bestimmung der Umgebungskräfte und -momente präsentiert. Anschließend wird eine Methode zur Bildverarbeitung vorgestellt, welche es erlaubt, den Zustand manipulierter Objekte mit hoher Geschwindigkeit zu verfolgen und zu präzisieren. Die beiden Module werden, zusammen mit der Aktionsplanung für den Roboter, in ein regelungstechnisches Rahmenwerk integriert. Zur Verbesserung der Systemperformanz wird der Einsatz eines Endeffektors mit intrinsischer Nachgiebigkeit evaluiert. Die Leistungsfähigkeit der entwickelten Lösungsmethoden sowie des gesamten Rahmenwerks wird in zahlreichen Simulationen und Experimenten demonstriert.

1 Introduction

For more than 30 years, robots have been extensively used in various industrial settings. There, they perform a broad range of manufacturing tasks with high speed and high accuracy. Typically, the precision is achieved by using a rigid mechanical design. Also, the robots operate in structured and known environments which facilitates the task execution and reduces the sensor requirements.

In spite of the great achievements in robotics over the last decades, the evolution of the field is still near the beginning. The potential areas of application for robots have gradually extended beyond the classical industrial settings in large-scale enterprises: lower costs and increased capabilities have increased the adoption of robots in smaller enterprises with flexible manufacturing systems [1]. Besides, service robots have gained growing attention in fields such as health care and even domestic services. Nowadays, the integration of robots into daily life has become a central development in robotics. Such an integration poses a fundamental challenge as both, the general conditions and the requirements for the robot, are drastically different from industrial applications. Robots that are destined to leave the classical industrial settings need to have various additional skills and capabilities compared to their counterparts in industrial settings. Fig. 1.1 illustrates the evolution of robotics over the last fifty years: the first industrial robot from Unimate, a six DOF industrial robot from ABB, and Honda's Asimo, which is currently one of the most advanced humanoid robots.

For the envisioned application areas mentioned above, a high degree of autonomy constitutes a characteristic feature. Tasks or plans are specified on a high-level and ultimately, the robot should be able to decide by itself which actions to take. This is a fundamental difference to classical industrial applications where robots typically execute predefined trajectories. Additionally, the assumption of a static environment does no longer hold for the new areas of application. The robot has to operate in changing and unknown surroundings. This necessitates an extensive environment perception and substantially increases the sensor requirements. It is crucial for the robot to rely on different sensor modalities and to fuse the information obtained from these sources. In addition to these sensor skills, advanced manipulation capabilities are needed. For the physical interaction with an unknown and changing environment, autonomous robots need a higher manipulative dexterity than their industrial counterparts. Also, the robot is no longer stationary but is requested to move in its environment. Such locomotion capabilities are typically realized either through bipedal walking or through a wheeled platform. This leads to challenging problems with respect to navigation and path planning. A number of other challenges are related to human-robot interaction: first and foremost, in order to become a valued partner, it is a prerequisite that the robot does not unsettle or scare the human by its appearance. The integration of robots into human life also implies physical interaction. Therefore, the second challenge with respect to human-robot interaction is the aspect of



Fig. 1.1: The first industrial robot Unimate (left), the six DOF industrial robot IRB 260 from ABB (middle) and Honda’s humanoid robot Asimo (right).

safety and dependability. While malfunctions in industrial applications are primarily a financial issue, they can have severe consequences if they occur during an interaction with a human. Hence, the creation of intrinsically safe systems is a fundamental goal for the mechanical design of next generation robots.

Concluding the previous considerations, it becomes apparent that the integration of robots into human life requires a completely new kind of robot. All the aforementioned requirements can be summarized as the robots’ ability to interact with humans *on equal terms*. And while these aspects are of great importance for the development of cognitive systems that can truly interact with humans, the main focus of this thesis is on the manipulation capabilities of robots. With respect to manipulation, such an interaction on equal terms implies that robots need to have manipulation skills and action/reaction times comparable to those of humans: interaction with a robot that can manipulate objects by throwing, catching, rolling, or sliding motions is intuitive and natural for humans who often exploit the advantages of dynamic manipulation in their own actions.

1.1 Overview of Manipulation

The following section provides definitions used to identify and to characterize the challenges that are associated with dynamic object manipulation. To this end, the various manipulation techniques are classified and the differences between them are illustrated. Furthermore, it is necessary to define and characterize different grasping techniques. Typically, dynamic object manipulation follows a nonprehensile approach and uses only basic form- or force-closure grasps. Manipulation tasks are, in general, also characterized by varying contact situations which will also be addressed at the end of the section.

Manipulation. The term manipulation is defined in different context. In a psychological interpretation, manipulation describes a type of social influence that tries to change the perception and/or action of others through hidden activities [44]. In a technical interpre-

tation, manipulation has a variety of meanings which all refer to physical changes in the surrounding environment. These changes include moving one or multiple objects, joining two or more objects by welding or gluing, or reshaping objects by cutting or grinding [88]. As pointed out by Bicchi, manipulation skills are, together with speech, probably the most important feature that distinguish humans from animals [10].

Throughout this thesis, the term manipulation refers to the moving of objects. In robotics, such an object manipulation is realized by robotic hands or end effectors. The terms hand or end effector denote the interface between the robotic arm and the environment [92].

Comparing manipulation of humans and robots, Mason points out some fundamental differences between the two [88]. One important aspect is the fact that humans possess a larger number of (distributed) sensors and actuators. Additionally, in contrast to humans, robots do not have the intrinsic capability to adapt to a given task. Here, human help is needed, either by providing task instructions or by equipping the robot with learning capabilities.

For the successful execution of a manipulation task, decisions have to be made with respect to the explicit execution strategy. It applies to both systems, human and robotic, that some of these decisions are made online while others are made offline, before the task is executed. In principle, humans have a much higher intrinsic ability to make decisions online [88]. Besides offline and online decisions, Mason differentiates a third type of decision which he labels off-offline. Such decisions refer to the design stage of the system. Typically, this includes the degrees of freedom, the sensor configuration, and the amount of actuation of the robotic system.

With respect to the aforementioned decision types, the theory of manipulation distinguishes two challenges which are obviously coupled: the first challenge are task-related, offline and online decisions which are needed for successful task execution. The second challenge are decisions with respect to the mechanical design of the robotic manipulators which are destined to perform the manipulation tasks.

Taxonomy of manipulation techniques. The following taxonomy of manipulation is adopted from Mason and Lynch [90]. As illustrated in Tab. 1.1, they distinguish four classes of manipulation based on the elements which are needed for a complete description. The classification is based on the terms kinematics, statics, and dynamics. Kinematics refers to the motion of bodies without considering the forces/torques that cause the motion [125]. It can be regarded as the geometry of the motion. Statics, in contrast, deals with the analysis of forces/torques in mechanical (physical) systems that are in a static equilibrium. Finally, dynamics is concerned with the forces/torques acting on bodies and the motions that are related to these forces/torques. Based on the three definitions, the following manipulation techniques can be characterized [88]:

- Kinematic manipulation: an action (or sequence of actions) which can be fully analyzed on the kinematics level
- Static manipulation: an action (or sequence of actions) where considerations with respect to both, statics and kinematics, are needed for the analysis

Class	Kinematic manipulation	Static manipulation	Quasi-static manipulation	Dynamic manipulation
Kinematics	✓	✓	✓	✓
Static forces	-	✓	✓	✓
Quasi-static forces	-	-	✓	✓
Acceleration forces	-	-	-	✓

Tab. 1.1: Taxonomy of manipulation by elements needed for analysis, from [90].

- Quasi-static manipulation: an action (or sequence of actions) where frictional and/or impact forces are predominant and inertial forces can be neglected in the analysis
- Dynamic manipulation: an action (or sequence of actions) where inertial/acceleration forces and torques are essential elements and are hence needed for planning and analysis

In order to clarify the taxonomy, some examples are helpful: In a typical pick-and-place task, the goal is to move an object from an initial to a final configuration. Neglecting the grasp/release of the object, the fundamental problem is the planning of a collision-free path from an initial configuration A to a final configuration B. This is an example for a kinematic manipulation. Clearly, the process of *picking* and *placing* the object generally involves statics, since a stable equilibrium configuration has to be ensured when the object is grasped. Consequently, these operations are examples for static manipulation. Sliding an object over a surface is an example of a quasi-static manipulation. Here, the influence of inertial forces is negligible in comparison to the friction forces. Finally, throwing, catching, and juggling of objects are examples of dynamic manipulation tasks. As pointed out by Mason, most dynamic manipulation tasks also constitute kinematic and/or quasi-static phases. A throwing motion, for instance, typically begins with a form- or force-closure grasp before the object's degrees of freedom are sequentially released. From this consideration, Mason introduced a further differentiation of dynamic manipulation [88]: based on the length of the dynamic manipulation phases, tasks are classified into quasidynamic, briefly dynamic, and continuously dynamic tasks.

Considering the aforementioned examples, two classes of manipulation tasks can be identified: the first class includes tasks that can be carried out by different manipulation techniques. Bringing an object from an initial to a final configuration, for instance, can generally be realized with each of the four techniques. The second class of tasks are *intrinsically* dynamic and hence require the dynamic manipulation technique. Juggling, throwing, and dribbling are examples for this second class. Mason summarizes these aspects by referring the term dynamic manipulation to methods which actively exploit the task dynamics instead of merely tolerating them [90].

Dexterous manipulation. In the general context of robotics, dexterity is commonly understood as the capability of the robotic system to autonomously perform tasks with a certain level of complexity [125]. This is a rather broad definition of the term which leaves room for different interpretations.

With respect to manipulation, the widely accepted meaning of the term dexterous is the capability of a hand / end effector to relocate objects in an arbitrary way according to the requirements of a given task [8]. In addition, Okamura defines manipulations as dexterous in which multiple manipulators, or fingers, cooperate to grasp and manipulate objects [95]. This definition of dexterous manipulation can be regarded as a *kinematic / static* dexterity and has an interesting implication: typically, dexterous kinematic / static manipulation requires robotic hands with numerous DOF as it describes the reconfiguration of arbitrary objects with different shapes, surfaces, sizes, and weights. Consequently, Murray attests robotic end effectors / hands with few DOF clumsiness and a lack of dexterity [92].

However, apart from this kinematic dexterity, a second interpretation of the term dexterous manipulation exists which can be entitled *dynamic* dexterity [52, 106]. Sakaguchi describes the ability of a robot to perform tasks that require a dynamical interaction with the environment with this term [113]. Similarly, Burridge defines it as the robots ability to perform work on the environment by changing its kinetic and potential energy [23].

With these definitions, it becomes evident that a significant difference between kinematic and dynamic dexterity exists: in contrast to kinematic dexterity, a multi-fingered hand or an end effector with multiple DOF is not an essential feature for a dynamically dexterous robot. For various tasks, a hand with multiple DOF has no or only little advantage over a basic end effector structure. Numerous case studies realized this type of dexterity with rather simplistic end effector design [89, 85, 8]. Clearly, this does not mean that sophisticated hands and dynamic dexterity exclude each other. However, it shows that there are different ways to manipulative dexterity: putting an increased effort into the task planning stage might enable the use of less complex and less expensive end effectors. Lynch and Mason demonstrated impressive manipulation skills with a minimalistic gripper design [82, 83, 84]. Bicchi also addressed this aspect by considering rolling contacts as a way to enhance manipulative dexterity [11].

The focus of this thesis is on planning and control methods for tasks which require robots that possess dynamic dexterity. Also, the goal is to realize these tasks with generic and hence simple end effector designs. While this reduces the costs of the required end effectors, it also constrains the variety of grasping operations that can be performed by the robot. To understand the concepts in the following chapters, a classification of grasping techniques is helpful.

Grasping. For grasping an object, the fundamental requirement is the ability to fix the object in an arbitrary pose (meaning position and orientation) relative to the hand [125, 9]. A characterization of grasps commonly distinguishes between form and force closure. These terms originate from the mechanical engineering domain where they are used to characterize mechanical joints.

With a *form closure* grasp, any infinitesimal motion of the object relative to the hand leads to a penetration of the two bodies. Hence, external forces/torques applied to the object result in normal forces on the contacting surfaces which counteract and balance the external forces.

With a *force closure* grasp, in contrast, a motion of the object relative to the hand does not necessarily result in a penetration of the two bodies. A force closure grasp only

requires that grasp forces exist which counteract the external forces/torques applied to the object. Typically, this fixation of the object is realized by normal forces at the contact points which, in turn, create friction forces.

Consequently, these definitions imply that a form closure grasp is also a force closure grasp since it also creates forces that balance the external forces applied to the object. The fundamental difference is that force closure allows frictional forces for the fixture of the object. These are tangential to the contacting surfaces, whereas the counteracting forces of a form closure grasp are normal to the contacting surfaces.

Clearly, a mixture of these grasping types is also possible and commonly referred to as *partial form closure*. Here, some DOF of the object are constrained by form closure and some by force closure.

Mason and Lynch applied their taxonomy of manipulation to grasping operations [90]. Based on the aforementioned definitions, a form closure grasp corresponds to a kinematic grasping operation since the grasp is determined by the joint configuration of the hand. Accordingly, a force closure is equivalent to a static grasping manipulation as frictional forces are essential for maintaining the grip. However, there are two additional manipulation techniques that are not covered by the definitions of form and force closure: quasi-static and dynamic manipulation. An example for the former is an end effector that pushes an object across a surface. Here, the contact between end effector and object represents a quasi-static grasp since the forces that maintain the contact between hand and object are created by sliding friction. Finally, a dynamic grasp uses acceleration forces to maintain the contact between hand and object. An example is the dribbling of a ball, where contact is only maintained as long as the downward acceleration of the hand is larger than the gravitational acceleration. Such a grasp is also referred to as *dynamic closure*.

Nonprehensile manipulation. The term nonprehensile manipulation denotes operations that are performed without form or force closure grasps [36]. Nonprehensile manipulation is sometimes entitled as graspless [89]. However, based on the aforementioned grasping definitions, operations such as dynamic closures or quasi-static grasps are also nonprehensile. Hence, it seems more appropriate to denote it as *manipulation without form- or force closure grasps* instead of graspless manipulation. Comparing it with human manipulation, it can be considered as just using the palm of one hand.

1.2 Applications

Exploiting dynamics in manipulation can render the object handling faster, since complex grasping is avoided, more versatile, since robots can also cope with intrinsically dynamic environments, and cheaper, since end effectors are generally less complex and robots can be constructed to be less powerful. This extends the capabilities of conventional static manipulation and opens up a new range of applications previously not feasible for robots. In particular, the following applications can be envisioned:

- With conventional manipulation techniques, it is not possible to manipulate objects that are oversized for conventional grippers. The paradigm of dynamic manipulation

addresses this shortcoming: by using nonprehensile manipulation with rolling and sliding motions, the *handling of bulky objects* can be realized.

- The grasping of unknown objects with a force closure grasp poses safety concerns, e.g. if the stiffness properties of the object are not known. Hence, for *manipulating unknown objects*, it is useful to employ nonprehensile techniques.
- Dynamic manipulation *increases the flexibility* of transportation systems and logistic chains [37]. Manipulating objects by sliding, throwing, or catching can reduce the transit time of such systems and, at the same time, lower the costs.
- Dynamic manipulation skills are also beneficial for autonomous robots as they increase their manipulative dexterity. This, in turn, improves the *handling of unforeseen situations*, e.g. the robot can catch objects that fall from a table or out of a cupboard that is being opened. Furthermore, dynamic manipulation skills are useful for everyday manipulation tasks such as opening and closing doors. Here, humans typically employ an approach that is very similar to throwing and catching of objects.

1.3 Control Framework and General Approach

In order to equip robots with dynamic manipulation skills, an extensive control framework is needed. Such a framework poses challenges that are associated with environment perception, action planning and motion & interaction control. Fig. 1.2 depicts the overall control structure employed in this work. It consists of four main elements: robot action planning, dynamic force/torque observer, object tracking & trajectory prediction, and motion & interaction control.

The *robot action planning* discusses how to find optimal trajectories: it proposes selection criteria and cost functions for dynamic manipulation tasks. The developed methods

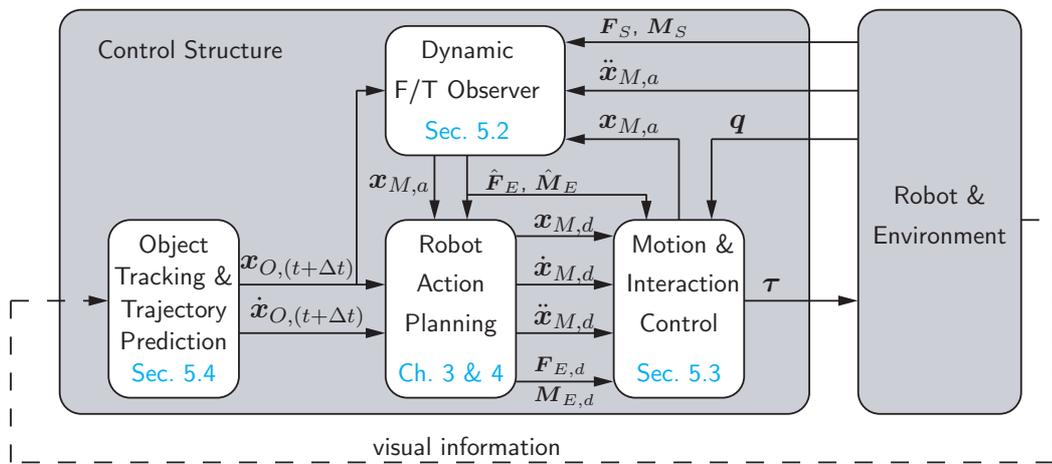


Fig. 1.2: Overall control structure consisting of four main elements: robot action planning, dynamic F/T observer, object tracking & trajectory prediction, and motion & interaction control.



Fig. 1.3: Stäubli six DOF industrial robot (left) and the dual-arm robot with 14 DOF (right).

are then applied to a number of case studies. Realizing these tasks on real hardware generally requires a perception of the environment. Here, the control framework considers visual and force/torque sensor information: the former is used for *object tracking* and the latter is used in a *dynamic force/torque observer* to reconstruct the interaction forces and torques. For both modalities, the use in dynamic manipulation tasks poses additional challenges. The type and/or the amount of feedback is chosen on a task-dependent basis. Furthermore, the execution of these tasks with real robots require precise *motion and interaction control* schemes. The control framework and its aforementioned components are evaluated in a number of experiments with the two robots depicted in Fig. 1.3.

Clearly, the hardware design is a crucial aspect for the planning and execution of dynamic manipulation tasks. While the mechanical design of the robotic manipulators is not within the scope of this work, design modifications of the end effector are considered. More specific, the thesis investigates how the use of elastic elements can enhance the robots performance in dynamic manipulation tasks.

The planning and control methods discussed in this work rely on models of the system and its environment. In contrast to the pursued model-based approach, learning strategies are a way to transfer the modeling effort from the human to the robotic system [90]. Both approaches have been successfully applied to a variety of manipulation tasks. However, from the authors' point of view, it is desirable to provide a task model whenever it is possible. Furthermore, also with model-based control, the system is capable of reacting on changing environment conditions based on adaptive control schemes.

The thesis considers the realization of dynamic manipulation tasks with generic end effector designs. However, it is important to point out that dynamic dexterity and multi-fingered hands do not exclude each other. On the contrary, dynamic manipulation tasks that require hands with multiple DOF do exist, e.g. twisting a pen in one's hand. Additionally, multi-fingered hands also fulfill other functions in addition to manipulation such as the exploration of objects [10]. Still, most dynamic manipulation tasks can be realized with simple end effector designs. Consequently, using such an approach is desirable as it generalizes to more complex hands, e.g. by using only the palm of a multi-fingered hand for a particular task.

1.4 Contributions and Outline of this Thesis

The following contributions are presented in this thesis:

Dexterous manipulation with generic end effector designs. The thesis illustrates how to equip robots with dexterous manipulation skills using simplified and generic end effector designs. It introduces the relevant modeling foundations, discusses optimization criteria for trajectory planning, and presents applications to validate the approach.

Control framework for dynamic object manipulation. The particular challenges for the closed-loop control of dynamic manipulation tasks are addressed: four different observer designs are developed to reconstruct the forces/torques exchanged between the robot and its environment. The designs are evaluated in a simulation scenario as well as in experiments. The force/torque observer is integrated in a direct force control scheme to improve the interaction control for dynamic motions. Finally, an approach for high-speed image processing is presented which allows to track and to predict the state of objects during non-contact phases with high sampling rates.

A novel case study for dynamic dexterity. Ball dribbling is introduced as a new case study for dynamic object manipulation. The system dynamics and optimal trajectory planning are discussed for both, a rigid and a compliant end effector design. In addition, the classic juggling task is considered: the thesis reviews the control strategies proposed in literature and presents an approach for optimal trajectory planning based on a non-local stability analysis. The thesis provides a comparison between the classic juggling and the novel dribbling task and elaborates why the latter is more challenging.

End effector design with intrinsic compliance. The thesis proposes the use of intrinsically compliant end effectors to improve the performance in dynamic manipulation tasks. The benefits of the approach are demonstrated in experiments which compare the performance of the compliant and the rigid end effector design for the dribbling task.

Experimental evaluation. Nonprehensile catching, throwing, dribbling, and juggling tasks are experimentally studied with a six DOF industrial robot and a 14 DOF anthropomorphic dual-arm manipulator. The experiments validate the approach of the thesis by integrating the results with respect to modeling, environment perception, planning and control design in a robotic basketball scenario. In addition, the experiments provide valuable insights into practical issues that have not been considered in the theoretical analysis.

The thesis is structured into four main parts: modeling foundations, action planning, perception & interaction control, and experimental evaluation. The outline is as follows: Chapter 2 presents the modeling foundations that are needed for the planning and execution of dynamic manipulation tasks. Then, Chapters 3 and 4 address the challenge of optimal motion and interaction planning for these tasks. The former chapter focuses on

non-periodic manipulation tasks that are intrinsically dynamic, whereas the latter chapter discusses periodic manipulation tasks with intermittent contact. Next, Chapter 5 discusses environment perception & interaction control for dynamic manipulation tasks. Finally, Chapter 6 presents a number of case studies in a robotic basketball scenario to evaluate the implemented control framework in experiments.

2 Modeling Foundations for Dynamic Object Manipulation

Summary The goal of this chapter is to summarize and to review state-of-the-art models that provide the foundations for dynamic object manipulation: a hybrid system model is introduced to capture the task dynamics with varying contact states. The contact kinematics are outlined in a general form and it is shown how the equations simplify for special cases such as rolling or sliding motions. In addition, the chapter discusses models for friction and for impact events.

As detailed in Chapter 1, dynamic object manipulation extends the classical manipulation techniques: besides (quasi-)static forces, it also involves acceleration forces. In addition, the relative motion between the robotic hand and the object is actively used. Examples for such a relative motion are rolling, sliding, spinning, and free flight phases with no contact. Due to relative motion, impact events and friction forces are central elements of dynamic object manipulation and must be included in the analysis. Consequently, non-smooth dynamics are an important aspect in modeling. Another consequence of the varying contact states are hybrid system dynamics. This necessitates a suitable mathematical model to describe the hybrid control system.

The outline of the chapter is as follows: first, Section 2.1 gives some preliminary definitions. Next, Section 2.2 details the modeling approach for discrete-continuous control systems that is used in this work. Then, Section 2.3 presents a classification of kinematic constraints that occur during contact. Section 2.4 outlines a mathematical model for the kinematics of two contacting bodies, considering different contact situations such as rolling, sliding, spinning, and free-flight (no contact). Then, Section 2.5 discusses friction models considering both, static and dynamic, modeling approaches. While these models are essential for task modeling, they are also important for precise robot motion control based on inverse dynamics. Finally, Section 2.6 focuses on the modeling of impact events. Classification criteria for impacts are outlined and two types of impact models are distinguished: discrete (impulse-momentum) and continuous (force-based) approaches.

2.1 Preliminaries

The section introduces some preliminary definitions. Detailed discussions of these terms can be found in the books by Khalil or Parker [99, 69].

Definition 2.1 (Vector field) A vector field on \mathbb{R}^m is a smooth map which assigns a tangent vector $\mathbf{f}(\mathbf{x}) \in T_q\mathbb{R}^m$ to each point $\mathbf{x} \in \mathbb{R}^m$. In local coordinates, \mathbf{f} is expressed as a column vector

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}. \quad (2.1)$$

Definition 2.2 (Autonomous ordinary differential equation) An n -th order autonomous ordinary differential equation is defined by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \quad (2.2)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the system state and $\mathbf{f} : D \rightarrow \mathbb{R}^n$ is a locally Lipschitz vector field (map) from domain $D \subseteq \mathbb{R}^n$ into \mathbb{R}^n . A unique solution exists for every initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. This solution is denoted by the flow $\phi_t(\mathbf{x}_0)$ which assigns a trajectory $\mathbf{x}(t)$ to every initial value \mathbf{x}_0 . \square

Generalized coordinates, configuration. A set of generalized coordinates

$$\mathbf{q}(t) = \begin{bmatrix} q_1(t) \\ \vdots \\ q_n(t) \end{bmatrix} \in \mathbf{Q} \subseteq \mathbb{R}^n \quad (2.3)$$

is used to describe the configuration of the system. The configuration space \mathbf{Q} is the space that contains all configurations of a given system.

Degrees of freedom. The degrees of freedom (DOF) of a system are defined by the number of independent generalized coordinates.

2.2 Hybrid System Model

One of the characteristic features of dynamic object manipulation are varying contact states. This necessitates a modeling framework for hybrid control systems. Various models for such systems have been proposed in literature, see e.g. [144, 17, 16, 25].

The hybrid system model (HSM) used in this work has been proposed by Buss [25] and is similar to the one developed by Branicky et al. [16]. At first sight, the model by Branicky seems to be more general as it explicitly allows state vectors with varying dimensions. However, this property is also implicitly included in the model by Buss, compare [25]. A further difference between the two models exists with respect to the definition of switching surfaces. Here, the model by Buss provides more flexibility as it allows a time dependence

of the switching surfaces. The following section outlines the hybrid system model used in this work. A detailed description of the HSM as well as a comparison with other modeling techniques for hybrid systems can be found in [25].

The state vector of the hybrid system is defined as

$$\boldsymbol{\zeta}(t) = \boldsymbol{\zeta} = \begin{bmatrix} \mathbf{x}(t) \\ x_d(t) \end{bmatrix} \in \mathbb{R}^n \times \mathbb{Z} \quad (2.4)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the continuous and $x_d(t) = 1, 2, \dots, n_d \in \mathbb{Z}$ the discrete state of the system. Accordingly, the output of a hybrid system comprises a continuous $\mathbf{y}(t) = \mathbf{h}(\mathbf{x}, \mathbf{u}, x_d, u_d, t)$ and a discrete part $y_d(t) \in \mathbb{Z}$. The continuous control input of the hybrid system is defined as

$$\mathbf{u}(t) = \mathbf{u} = \sum_{k=1}^{n_d} \delta_{k,x_d} \mathbf{u}_k(t) \in \mathbb{R}^m \quad (2.5)$$

with the Kronecker delta δ_{k,x_d} being 1 if $k = x_d$ and 0 if $k \neq x_d$. The discrete control input is given by $u_d(t) \in \mathbb{Z}$. For a compact notation, the time dependency of the variables is omitted in the following. With (2.4) and (2.5), the continuous dynamics are described by vector fields of the form

$$\mathbf{f} = \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{u}, x_d, u_d, t) = \mathbf{f}_1(\mathbf{x}, \mathbf{u}_1, x_d, u_d, t) & \text{for } x_d = 1 \\ \vdots \\ \mathbf{f}_{n_d}(\mathbf{x}, \mathbf{u}, x_d, u_d, t) = \mathbf{f}_{n_d}(\mathbf{x}, \mathbf{u}_{n_d}, x_d, u_d, t) & \text{for } x_d = n_d. \end{cases} \quad (2.6)$$

The occurrence of discrete events is defined by switching surfaces S_i^j

$$S_i^j : s_i^j(\mathbf{x}, \mathbf{u}, x_d, u_d, t) = 0. \quad (2.7)$$

If one of the switching conditions is fulfilled ($s_i^j(\mathbf{x}, \mathbf{u}, x_d, u_d, t) = 0$), the state of the system is reset (reinitialized) based on the transition or jump map

$$\boldsymbol{\zeta}^+ = \begin{bmatrix} \mathbf{x}^+ \\ x_d^+ \end{bmatrix} = \begin{bmatrix} \Delta_i^j(\mathbf{x}, \mathbf{u}_i, x_d, u_d, t^-) \\ j \end{bmatrix}. \quad (2.8)$$

Here, t^- denotes the limit from the left of time t and $\boldsymbol{\zeta}^+ = \boldsymbol{\zeta}(t^+)$ is the hybrid state immediately after the impact event (limit from the right). With (2.4)-(2.8), the definition of a hybrid control system is complete.

Definition 2.3 (Hybrid control system) A hybrid control system is defined by its continuous dynamics

$$\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, \mathbf{u}_i, x_d, u_d, t) \quad \text{if } s_i^j(\mathbf{x}, \mathbf{u}, x_d, u_d, t) \notin 0 \quad \forall \{i, j\}, \quad (2.9)$$

transition or jump maps

$$\boldsymbol{\zeta}^+ = \begin{bmatrix} \mathbf{x}^+ \\ x_d^+ \end{bmatrix} = \begin{bmatrix} \Delta_i^j(\mathbf{x}, \mathbf{u}_i, x_d, u_d, t^-) \\ j \end{bmatrix} \quad \text{if } s_i^j(\mathbf{x}, \mathbf{u}, x_d, u_d, t^-) = 0, \quad (2.10)$$

and outputs $\mathbf{y}(t) = \mathbf{h}(\mathbf{x}, \mathbf{u}, x_d, u_d, t)$ and $y_d(t)$. \square

2.3 Kinematic Constraints

Manipulation requires contact phases between object and hand or end effector respectively. Typically, a contact is modeled as kinematic constraint [92, 88, 126]. By imposing such a kinematic constraint on a mechanical system, the possible paths that the system can follow are reduced and the motion of the system is limited.

If contact is established by form or force closure grasp, object motion relative to the hand is prohibited and the object is immobilized. During nonprehensile manipulation, however, motion of the object is only constrained in some directions. The following summarizes the different types of constraints that can be identified. Further information can be found in [88, 92].

In general, bilateral and unilateral constraints are distinguished. The former are two-sided constraints that can be written as an equality $c(\dots) = 0$, the latter are one-sided constraints formalized as inequalities $c(\dots) \geq 0$. A further differentiation is made with respect to time dependency: a scleronomic constraint is stationary and hence time-independent whereas a rheonomic constraint is moving and time-dependent.

Holonomic constraint. As detailed by the following definition, a holonomic constraint can be expressed using only configuration variables and potentially time (as it can either be scleronomic or rheonomic).

Definition 2.4 (Holonomic constraint) A set of functions $c_{h,i}(\mathbf{q}, t) : \mathcal{Q} \times \mathbb{R} \rightarrow \mathbb{R}$ on the configuration space that can be written in the form

$$c_{h,i}(\mathbf{q}, t) = 0, \quad i = 1, \dots, k. \quad (2.11)$$

is called a holonomic constraint. They define a smooth hypersurface in the configuration space. \square

Nonholonomic constraint. Commonly, constraints that do not satisfy (2.11) are denoted nonholonomic constraints, compare the discussion in [88]. Based on this interpretation, all unilateral constraints are nonholonomic as they are not described with an equality.

Definition 2.5 (Nonholonomic constraint) A set of functions $c_{nh,i}(\mathbf{q}, \dot{\mathbf{q}}, t) : \mathcal{Q} \times \mathcal{Q} \times \mathbb{R} \rightarrow \mathbb{R}$ on the configuration space that can be written in the form

$$c_{nh,i}(\mathbf{q}, \dot{\mathbf{q}}, t) = 0, \quad i = 1, \dots, k. \quad (2.12)$$

is called a nonholonomic constraint. \square

Definition 2.6 (Pfaffian constraint) A constraint of the form

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (2.13)$$

is called a Pfaffian constraint. Hence, a Pfaffian constraint is a special case of a nonholonomic constraint. A Pfaffian constraint is integrable if a vector-valued function $\mathbf{c}_h(\mathbf{q}, t) : \mathcal{Q} \rightarrow \mathbb{R}^k$ exists such that $\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \Leftrightarrow \frac{\partial \mathbf{c}_h(\mathbf{q}, t)}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{0}$. \square

From this definition it follows that an integrable Pfaffian constraint is equivalent to a holonomic constraint [92].

Virtual holonomic constraint. Typically, the restrictions described by a holonomic constraint are physically imposed on the system. However, it is also possible to define artificial or virtual geometric constraints for a given system [123, 143]. In contrast to a holonomic constraint, these virtual holonomic constraints have to be fulfilled by some control action. As detailed in Chapter 4, such constraints can be used for the control of underactuated periodic manipulation tasks.

2.4 Contact Kinematics

Contact modeling is a fundamental aspect in the planning of robotic tasks. This includes robotic walking machines, robotic hands grasping objects, and part handling in industrial scenarios [86]. The following section considers contact kinematics from a *manipulation* perspective and outlines the model originally developed by Montana [91].

Assumptions. The derivation of the kinematic contact equations is based on the following assumptions:

- A1** Contacts can be regarded as *point* contacts.
- A2** For two bodies, contact occurs only at *one* point.
- A3** The surfaces of the contacting bodies are *regular*.

Surface parameterization. Given an object in \mathbb{R}^3 with an arbitrary regular surface S , this surface can be locally described by the orthogonal parameterization $\mathbf{c} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$,

$$\mathbf{c}(u, v) = \begin{bmatrix} f_1(u, v) \\ f_2(u, v) \\ f_3(u, v) \end{bmatrix}. \quad (2.14)$$

Tangent plane. The tangent plane to the surface is spanned by the vectors

$$\mathbf{c}_u(u, v) := \frac{\partial \mathbf{c}}{\partial u} \quad \mathbf{c}_v(u, v) := \frac{\partial \mathbf{c}}{\partial v}. \quad (2.15)$$

Kinematic equations of contact. In the following, the subscripts 1 and 2 denote the two bodies. A schematic of a general contact situation is depicted in Fig. 2.1. The surfaces of the two bodies in the body-fixed coordinate frames Σ_1, Σ_2 are parameterized according to (2.14). The contact frames Σ_{c1}, Σ_{c2} are defined as the normalized Gauss frames at the contact point. The angle ψ is defined as the rotation angle around the z -axis of Σ_{c2} that aligns the x -axes of Σ_{c1} and Σ_{c2} . Additionally, the body-fixed frames $\Sigma_{l1}(t), \Sigma_{l2}(t)$ are defined that coincide with the respective contact frame at time t . The motion of object 1

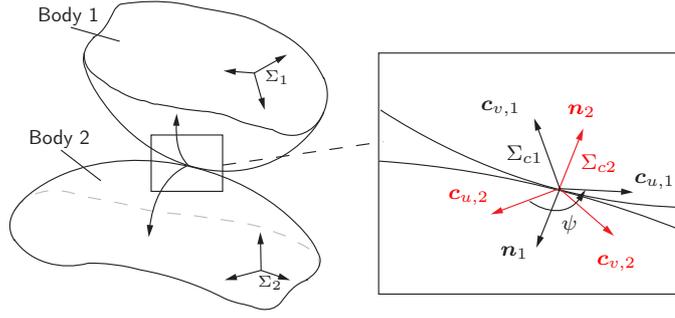


Fig. 2.1: Motion of two bodies in contact, adopted from [92].

relative to object 2 is then described by the translational $\mathbf{v} = [v_x \ v_y \ v_z]^T$ and rotational $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ velocity of $\Sigma_{l1}(t)$ relative to $\Sigma_{l2}(t)$. The contact kinematics are given by [91]

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \end{bmatrix} = \mathbf{M}_1^{-1} (\mathbf{K}_1 + \tilde{\mathbf{K}}_2)^{-1} \left(\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} - \tilde{\mathbf{K}}_2 \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right), \\ \begin{bmatrix} \dot{u}_2 \\ \dot{v}_2 \end{bmatrix} = \mathbf{M}_2^{-1} \mathbf{R}_\psi (\mathbf{K}_1 + \tilde{\mathbf{K}}_2)^{-1} \left(\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} + \mathbf{K}_1 \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right), \quad (2.16)$$

$$\dot{\psi} = \omega_z + \mathbf{T}_1 \mathbf{M}_1 \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \end{bmatrix} + \mathbf{T}_2 \mathbf{M}_2 \begin{bmatrix} \dot{u}_2 \\ \dot{v}_2 \end{bmatrix},$$

$$0 = v_z.$$

$$\text{with } \tilde{\mathbf{K}}_2 = \mathbf{R}_\psi \mathbf{K}_2 \mathbf{R}_\psi \quad \text{and} \quad \mathbf{R}_\psi = \begin{bmatrix} \cos \psi & -\sin \psi \\ -\sin \psi & -\cos \psi \end{bmatrix}.$$

Here, \mathbf{M} is the metric tensor, \mathbf{K} is the curvature tensor, and \mathbf{T} is the torsion form of the corresponding surface. The three quantities are commonly referred to as the geometric parameters of a surface. Details on these terms can be found in the book by Murray [92] and in Appendix A.1.

The equations (2.16) describe the kinematics of a point contact in its most general form. Special types of relative motion are realized by imposing additional constraints - this is detailed in the following subsections.

2.4.1 Sliding

A sliding motion imposes the constraint of zero rotational velocity,

$$\omega_x = \omega_y = \omega_z = 0. \quad (2.17)$$

2.4.2 Rolling

Here, *rolling* and *pure rolling* are distinguished. For the former, the following conditions

$$v_x = v_y = 0, \quad (2.18)$$

have to be fulfilled. *Pure rolling* adds the constraint of no spin motion,

$$v_x = v_y = \omega_z = 0. \quad (2.19)$$

2.4.3 Spinning

A spinning motion comprises only rotational velocities in normal direction to the tangential contact plane,

$$\omega_x = \omega_y = v_x = v_y = 0. \quad (2.20)$$

2.4.4 No Contact / Free-Flight

Finally, the special case of *no contact* between object and end effector is considered. Here, none of the objects' six DOF is constrained and nonzero velocities v_z are allowed [88]. The motion of the object is determined by gravitational acceleration and possibly air resistance.

2.5 Friction

Friction is a nonlinear phenomenon which is of great importance for various engineering disciplines. With respect to this work, there are three reasons for the interest in accurate friction models: first, nonprehensile manipulation includes relative motion between object and hand such as rolling or sliding. Hence, to plan these tasks, a model of the friction forces is essential. The second reason is related to the motion control of robotic manipulators. The successful execution of dynamic manipulation tasks with robotic systems requires a precise motion control of the robot. This is typically achieved by an inverse dynamics scheme which necessitates an accurate description of the friction forces as part of the dynamic model, see Sec. 5.3 for details. Finally, friction is an important aspect when considering impact between bodies with nonsmooth surfaces.

Numerous friction models have been proposed. Detailed surveys on these models can be found in [4, 97]. In general, one distinguishes static and dynamic friction models. The characteristic feature of static models is that the friction force only depends on the current velocity. Hence, non-stationary friction phenomena such as micro-slip or hysteretic behavior are not captured by static models. This shortcoming is addressed by dynamic (or state variable) friction models. These models introduce one (or more) state variable(s) and the friction force is a function of the current velocity and the state variable(s). In the following, static and dynamic friction models are briefly discussed and their suitability for the different purposes (trajectory planning, impact modeling, and friction compensation in motion control) is evaluated.

2.5.1 Static Models

This category includes all models that use *static* maps between velocity and friction force [27]. Classical static friction models comprise a combination of four components, which capture the fundamental friction phenomena.

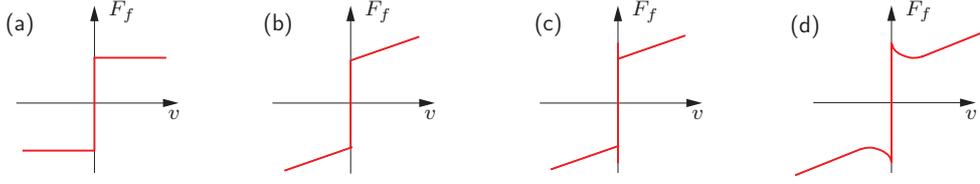


Fig. 2.2: Static friction models: (a) Coulomb friction, (b) Coulomb and viscous friction, (c) Coulomb, viscous, and static friction, (d) Coulomb and viscous friction with Stribeck effect.

Coulomb friction. The basic idea of Coulomb friction is that the friction force opposes the motion and that its magnitude is independent of velocity and contact area [97]. Hence, it is defined as

$$F_c(v, F_n) = \mu_c F_n \text{sgn}(v) = F_{c,0} \text{sgn}(v). \quad (2.21)$$

Here, $\text{sgn}(\cdot)$ denotes the signum function and v is the relative velocity of the two contact areas. $F_{c,0}$ is a force proportional to the normal load F_n given by $F_{c,0} = \mu_c F_n$ with the Coulomb friction coefficient μ_c , compare Fig. 2.2(a).

Viscous friction. The second friction component, viscous friction, describes the friction that is caused by the viscosity of lubricants. Hence, it is a function of velocity:

$$F_v(v) = k_v v \quad \text{or} \quad F_v(v) = k_v \|v\|^{\delta_v} \text{sgn}(v), \quad (2.22)$$

where k_v is the viscous friction coefficient, $\|v\|$ denotes the norm of v , and δ_v is a parameter that depends on the geometry of the application [97]. The combination of Coulomb and viscous friction is depicted in Fig. 2.2(b).

Static friction. Coulomb and viscous friction do not cover the static friction forces at zero velocity. These friction forces prevent relative motion by counteracting the external forces F_e up to a friction threshold or break-away force $F_{s,t} = \mu_s F_n$ determined by the normal force F_n and the static friction coefficient μ_s . Static friction is described as

$$F_s(F_n, F_e) = \begin{cases} F_e & \text{if } v = 0 \text{ and } \|F_e\| < F_{s,t} \\ F_{s,t} \text{sgn}(F_e) & \text{if } v = 0 \text{ and } \|F_e\| \geq F_{s,t}. \end{cases} \quad (2.23)$$

The combination of Coulomb, viscous, and static friction is depicted in Fig. 2.2(c).

Stribeck effect. The static friction components yields a discontinuous friction force at the transition from rest to relative motion. As this discontinuity contradicts experimental observations, Stribeck proposed a modification of the friction curve at non-zero velocities [135],

$$F(v, F_n) = (F_{s,t} - F_{c,0}) e^{-\|v/v_s\|^{\delta_s}} \text{sgn}(v), \quad (2.24)$$

with the empirical parameters v_s (Stribeck velocity) and δ_s (commonly $\delta_s = 2$). The combined model comprising the four components is depicted in Fig. 2.2(d).

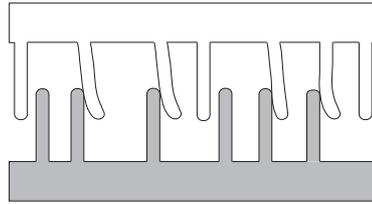


Fig. 2.3: Bristle model: representation of the friction phenomenon on a microscopic level, adopted from [48]. For simplicity, the bristles of the lower surface are shown rigid.

2.5.2 Dynamic Models

Static models fail to explain some additional friction phenomena that have been experimentally observed such as micro-slip, hysteresis, and varying break-away forces. Dynamic models address these shortcomings and describe friction at a microscopic level. At this scale, the surfaces are irregular and contact occurs only at certain points which are called asperities. Haessig et al. proposed to represent this contact state with a bristle model illustrated in Fig. 2.3. At various points, the elastic bristles of the two surfaces are in contact with each other and the relative motion of the bodies causes a deflection of the bristles. For each bristle, the deflection results in a force that contributes to the friction force, for details see [48]. Although the original bristle friction model from Haessig is computationally inefficient, the basic idea of modeling the contact with bristles has been used in various models.

Dahl model. Dahl's model was motivated by the observation of micro-slip, also called pre-sliding displacement or Dahl effect, between contacting surfaces [32]. The model captures the spring-like behavior of the bristle model, without explicitly using it. In its common form, the Dahl model is given by

$$\begin{aligned}\dot{z} &= v - \frac{\|v\|}{F_{c,0}} \sigma_0 z, \\ F &= \sigma_0 z.\end{aligned}\tag{2.25}$$

Here, the internal state variable z represents the average bristle deflection and σ_0 is the bristle stiffness. While the model captures the micro-slip phenomenon, it does not include the Stribeck effect.

LuGre model. The LuGre (Lund Grenoble) model was proposed by Canudas de Wit et al. in [27]:

$$\begin{aligned}\dot{z} &= v - \frac{\|v\|}{g(v)} z \\ F &= \sigma_0 z + \sigma_1(v) \dot{z} + k_v v\end{aligned}\tag{2.26}$$

with $g(v) = \frac{1}{\sigma_0} \left(F_{c,0} + (F_{s,t} - F_{c,0}) e^{-(v/v_s)^2} \right)$.

Again, the variables z and σ_0 denote the average bristle deflection and stiffness, $\sigma_1(v)$ is the damping, and $g(v)$ captures Coulomb friction and Stribeck effect. When $g(v) = F_c/\sigma_0$ and $\sigma_1 = k_v = 0$, the LuGre model reduces to the Dahl model. The central advantages of the LuGre model are its completeness with respect to the modeled friction phenomena and the relatively low number of parameters. In total, six parameters have to be identified for the LuGre model: the ones already known from the static models ($F_{c,0}$, $F_{s,t}$, k_v , v_s) and the two additional parameters σ_0 and σ_1 .

Beside these two, other dynamic friction models have been proposed: Bliman and Sorine presented an extension of Dahl's model using two state variables [13]. However, their model does neither capture the Stribeck effect nor frictional lag [40]. Armstrong-Hélouvry introduced a dynamic friction model with seven parameters. This model includes the various friction phenomena by using different models for stiction and sliding [4].

2.6 Impacts

For interacting bodies, impacts are an essential phenomenon: they occur when two or more bodies collide with each other and are characterized by a short time duration and high force levels. For various research areas, such impact events are of great relevance. Popular examples are walking/hopping machines or drive trains with backlash. Impacts are also an important element in dynamic manipulation. To this end, the section characterizes impact types and reviews the state-of-the-art models to describe them. The aim of these models is to determine/predict the post-impact state of the system based on its pre-impact state. Clearly, the modeling process depends on the intended usage of the model: an online usage imposes constraints on the model complexity and hence reduces the accuracy.

The following subsections discuss the impact events that are relevant for dynamic object manipulation, namely collisions between two bodies with a single contact point. A general introduction to non-smooth dynamics can be found in the book by Brogliato [18]. An extensive literature survey on the modeling of impact dynamics was provided by Gilardi and Sharf [45].

2.6.1 Classification Criteria

When an impact event occurs, the two colliding bodies get in contact. The case of a general impact between two bodies is illustrated in Fig. 2.4. The line of impact is the common normal on the contacting surfaces. In the following, velocities along the line of impact are denoted as normal velocities with subscript n . Accordingly, velocities in the tangential plane of the contact are labeled with the subscript t .

Independent of the model that is used for the mathematical description, impacts are classified based on the following criteria [45].

Direct and oblique impact. As depicted in Fig. 2.5(a), an impact is denoted as *direct* if the pre-impact velocities of the two bodies are along the line of impact. Accordingly, an *oblique* impact occurs if one or both initial velocities do not fulfill this condition, compare Fig. 2.5(b). In the latter case, at least one body has a non-zero tangential velocity.

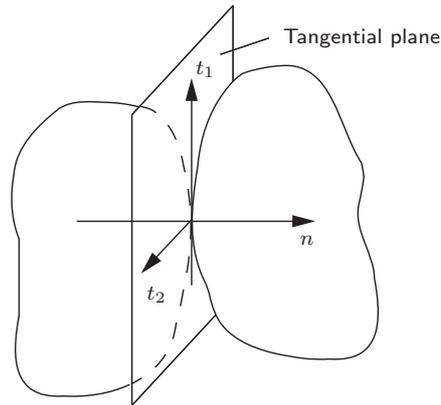


Fig. 2.4: Illustration of a general impact between two bodies.

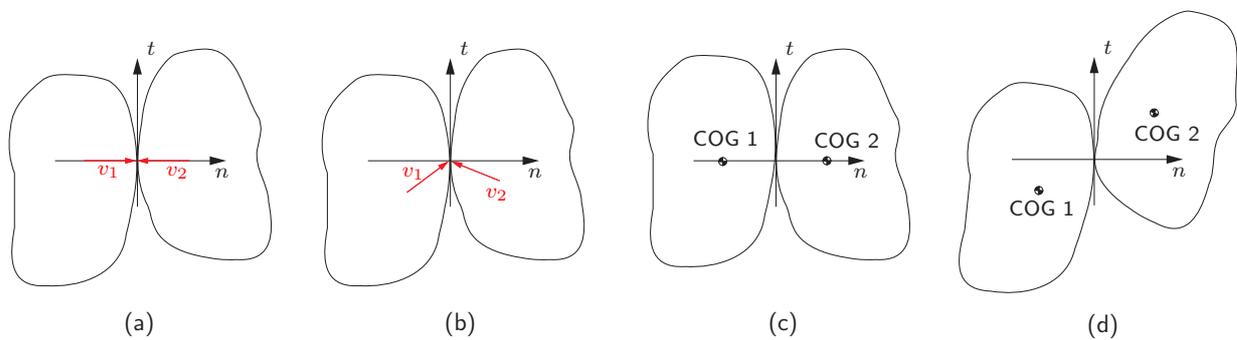


Fig. 2.5: Classification of impacts: (a) direct, (b) oblique, (c) central / collinear, (d) eccentric.

Central and eccentric impact. An impact is characterized as *central* or *collinear* if the centers of mass are on the line of impact, see Fig. 2.5(c). Consequently, the impact forces in normal direction do not change the rotational velocity. In contrast, for an *eccentric* impact the center of mass of one or both bodies is not on the line of impact, compare Fig. 2.5(d). Hence, the rotational velocity after the impact is changed due to the impact forces in normal direction.

Energy loss. For all impact events, a *compression* and a *restitution* phase can be distinguished. The compression phase begins with the initial contact of the two bodies. The transition to the restitution phase occurs at the time of maximum deformation when the relative velocity between the two bodies is zero. The end of the restitution phase is reached when the bodies separate.

Depending on the energy loss during the impact event, four impact types are characterized: a *perfectly elastic* impact occurs when no kinetic energy is lost during the collision and the deformation is completely reversed. In contrast, for a *perfectly inelastic* or *plastic* collision, the maximum possible amount of kinetic energy is lost, the two bodies deform and stick together after the impact. If the kinetic energy is not completely conserved but the deformation is completely reversed, the impact is *partially elastic*. Finally, an impact with a partial loss of kinetic energy and some amount of permanent deformation is denoted as *partially plastic*.

The first two types are idealized cases: a perfectly elastic impact assumes that the energy flow from the initial kinetic energy to the elastic strain energy is completely reversible. Similarly, a perfectly inelastic collision assumes that the lost kinetic energy is completely transferred into plastic deformation. In reality, impacts of two colliding masses are partially plastic or partially elastic and hence a mixture of the idealized cases. Typically, not all of the initial kinetic energy is transformed into elastic strain or plastic deformation: some energy is dissipated in other processes such as wave propagation, sound, and heat [45].

Smooth and frictional impact. Friction is an important aspect as it can stop or reverse the tangential motion during an impact. In contrast, if the impact is assumed frictionless, no tangential forces can be exchanged between the two bodies and the impulse in tangential direction, ΔP_t is zero.

2.6.2 Discrete Models

Discrete or impulse-momentum-based impact models are based on the following assumptions: the impact event is instantaneous and the occurring impact forces are impulsive. This implies discontinuous changes in the velocities of the colliding bodies while the positions are invariant. In addition, forces that are not caused by the impact (e.g. gravitational or spring forces) are assumed to be negligible. Typically, discrete models are used for the collision of bodies that can be considered as *effectively rigid* [45, 145]. This criteria is fulfilled if strain energy is negligible and the contact area is small compared to the size of the body [136]. Discrete models are based on the specification of a coefficient of restitution (COR) c_r , which describes the plasticity of the impact and varies in the range from 0 (perfectly plastic) to 1 (perfectly elastic). The aforementioned assumption of an instantaneous impact with discontinuous changes in the velocities results in infinitely large impact forces. Hence, it is more suitable to deal with impulses instead. In the following, the subscript specifies the body (1 or 2) and the superscript + denotes post-impact velocities.

The *change in momentum* over a time interval $[t_i, t_f]$ or the so called *impulse-momentum equations* are given by

$$\begin{aligned}\Delta \mathbf{P} &= m\Delta \mathbf{v} = m(\mathbf{v}_f - \mathbf{v}_i) = \int_{t_i}^{t_f} \mathbf{F} dt, \\ \Delta \mathbf{L} &= \mathbf{J}\Delta \boldsymbol{\omega} = \mathbf{J}(\boldsymbol{\omega}_f - \boldsymbol{\omega}_i) = \int_{t_i}^{t_f} \mathbf{M} dt = \int_{t_i}^{t_f} \mathbf{r} \times \mathbf{F} dt,\end{aligned}\tag{2.27}$$

where \mathbf{P} / \mathbf{L} denotes the linear / angular momentum, m and \mathbf{J} are the mass and the inertia tensor of the body, and \mathbf{r} is the vector from the center of mass to the contact point. With conservation of momentum, the sum of momentum of two bodies before and after a collision is equal

$$m_1 \mathbf{v}_1 + m_2 \mathbf{v}_2 = m_1 \mathbf{v}_1^+ + m_2 \mathbf{v}_2^+, \tag{2.28}$$

$$\mathbf{J}_1 \boldsymbol{\omega}_1 + \mathbf{J}_2 \boldsymbol{\omega}_2 = \mathbf{J}_1 \boldsymbol{\omega}_1^+ + \mathbf{J}_2 \boldsymbol{\omega}_2^+. \tag{2.29}$$

Discrete models decouple the impact problem into a contact problem in normal direction

and a friction problem in tangential direction [46]: the linear impulse $\Delta\mathbf{P}$ is divided in a normal impulse ΔP_n and a tangential impulse ΔP_t . Accordingly, the relative translational velocity at the contact point C is divided into a normal component $v_{C,n}$ and a tangential component $v_{C,t}$.

Normal direction. In normal direction, the most common models are the impact laws by Poisson, Newton, and Stronge.

- **Poisson's COR.** The normal impulse ΔP_n of an impact is the sum of the impulses in the compression and restitution phase

$$\Delta P_n = \Delta P_{n,c} + \Delta P_{n,r}. \quad (2.30)$$

In Poisson's hypothesis, the coefficient of restitution is defined as the ratio of the compression and the restitution impulse

$$c_{r,P} = \frac{\Delta P_{n,c}}{\Delta P_{n,r}}. \quad (2.31)$$

- **Newton's COR.** While Poisson's COR is based on the impulses during impact, Newton's model determines the coefficient of restitution based on velocities. Let $v_{C,n}$ ($v_{C,n}^+$) denote the relative normal velocity of the contact point C before (after) the impact. Then, Newton's model defines the coefficient of restitution as

$$c_{r,N} = -\frac{v_{C,n}^+}{v_{C,n}}. \quad (2.32)$$

- **Stronge's COR.** Stronge's model defines the COR as ratio of the work W in restitution (subscript r) and compression phase (subscript c) [136],

$$c_{r,S} = -\frac{W_r}{W_c}. \quad (2.33)$$

Tangential direction. The aforementioned three coefficients are for the normal direction. However, if the contacting surfaces are not smooth, changes in the translational velocities tangential to the impact plane and the rotational velocities will be induced [45]. A common method to reflect these changes is related to Newton's COR and specifies coefficients of restitution for these directions as

$$c_{r,t} = -\frac{v_{C,t}^+}{v_{C,t}} \quad c_{r,r} = -\frac{\omega_C^+}{\omega_C}. \quad (2.34)$$

Brach proposed an alternative approach and related the tangential and normal impulse [15],

$$\Delta P_t = \mu \Delta P_n. \quad (2.35)$$

For collinear or smooth (non-frictional) impacts, the models of Poisson, Newton, and Stronge are equivalent. However, for frictional impacts with transitions from slip to stick

or slip-reversal, Newton’s and Poisson’s model can lead to different energetic inconsistencies [136]. Despite this shortcoming, velocity-based coefficients of restitution will be used in this work as they provide an accurate description for central impacts of effectively rigid bodies and as their mathematical compactness allows a direct integration into the task planning.

2.6.3 Continuous Models

Continuous or force-based impact models relax the assumptions of the discrete models: they consider a finite impact duration and the positions of the colliding bodies are not fixed during impact. As the deformation of the objects is explicitly taken into account, continuous models are particularly well suited for the impact of elastic bodies [76]. Various continuous models have been proposed to capture the deformation and energy loss during impact, e.g. the models of Hertz, Kelvin-Voigt, and Hunt-Crossley [50, 58]. Commonly, continuous models include (possibly nonlinear) *spring* and *dampers* forces. In the following paragraph, this is exemplarily illustrated for the Hunt-Crossley model.

Hunt-Crossley model. For this model, the impact force in normal direction consists of a nonlinear spring and a nonlinear damping force

$$F(t) = -k_s x^{\lambda_1} - k_d x^{\lambda_2} \dot{x}. \quad (2.36)$$

where x is the penetration, \dot{x} the penetration velocity, k_s the spring constant, and k_d the damper constant [58]. The exponents λ_1, λ_2 take into account the geometry of the contact and in its common form, $\lambda = \lambda_1 = \lambda_2$ takes a value close to one. Methods for the parameter identification of continuous models have been proposed in [7, 47, 33].

2.6.4 Extensions

Multibody systems. The previous models considered the collision of two bodies at a single contact point. Impact situations for multi-body systems and/or with multiple contacts were addressed by Glocker and Pfeiffer [46, 100]. Studer investigated such systems of rigid bodies with non-smooth dynamics and developed a simulation environment [137].

Elastic bodies. Discrete and continuous model are both based on lumped parameters. For elastic bodies, distributed models can be used to describe the deformation and the energy loss during impacts. The Bullet physics project is one of the few examples that considers soft body dynamics [30]. Tian and Jia presented a distributed model for the specific class of deformable, shell-like objects [138].

2.7 Summary

This chapter provided the modeling foundations for dynamic object manipulation. To this end, state-of-the-art models for the relevant areas were outlined and briefly discussed.

Section 2.2 introduced a hybrid system model to cover the discrete and continuous aspects that are characteristic for dynamic manipulation tasks. Section 2.3 discussed the different types of kinematic constraints that occur during contact. Then, in Section 2.4, the kinematic equations for two contacting surfaces were derived. Next, Section 2.5 outlined the fundamental properties of static and dynamic friction models. For the planning of manipulation tasks, the static models provide sufficient accuracy. For friction compensation in robotic motion control schemes, however, dynamic models are generally preferable. Here, the LuGre model offers the best trade-off between accuracy and mathematical complexity. Finally, Section 2.6 considered the modeling of impact events. Both, discrete and continuous, impact models were outlined. The former are based on the impulse-momentum equations and assume an instantaneous impact. The mathematical compactness of discrete models comes at the cost of reduced accuracy and potential inconsistencies for frictional impacts. In contrast, the continuous models consider a finite impact duration and motion of the objects during impact. Naturally, this results in an increased mathematical complexity. As the manipulated objects in this work are considered to be effectively rigid, the discrete models provide an accurate description. For manipulation tasks with elastic bodies, continuous or distributed impact models are preferable.

3 Non-Periodic Dynamic Manipulation Tasks

Summary This chapter discusses non-periodic dynamic manipulation tasks and details how dynamic dexterity can be realized with generic end effector designs. The planning of optimal trajectories is addressed: the chapter summarizes the constraints that have to be considered and discusses criteria for the trajectory optimization. Three exemplary applications are studied: first, rolling manipulation is considered with a ball-on-plate system. Second, dual-handed ball throwing using a basic force closure grasp is addressed. Third, a *nonprehensile* approach for robotic catching is presented and two catching strategies are discussed.

While there are animals with locomotion or sensor capabilities superior to those of humans, the dexterous manipulation skills of humans are outstanding [10]. This is due to two reasons: superior grasping capabilities and a better coordination between sensor and motor skills. For *conventional* dexterous manipulation, advanced grasping capabilities are a prerequisite. This is illustrated by the fact that a general form closure grasp requires seven contact points [125]. Hence, multi-fingered hands are needed to manipulate objects of different size and shape with a form closure grasp. For *dynamical* dexterous manipulation, the grasping capabilities are less important. Many dynamic manipulation tasks can be performed in a nonprehensile fashion using only a single contact point between end effector and object [57, 84]. Instead, these tasks require a high ability of coordination between sensor and motor skills.

In general, the successful execution of manipulation tasks requires task-specific decisions. Some of these decisions can be made offline while others have to be made online during the task execution. Offline decisions commonly include a preselection that specifies the overall strategy for a particular task. Online decisions include the compensation of disturbances or the reaction to unforeseen events. The planning approach pursued in this work includes both decision types: first, a number of selection criteria are evaluated offline. The creation of look-up tables then allows to evaluate these selection criteria online and create the corresponding trajectory.

The outline of the chapter is as follows: first, Section 3.1 provides an overview of related work. In Section 3.2, the planning of motion and force trajectories for these tasks is addressed. Then, Section 3.3 presents applications for rigid end effector designs.

3.1 Related Work

Robots with dynamic skills have been studied in various research works: Hodgins and Raibert were among the first and investigated dynamic locomotion primitives such as hopping for robots [52]. With respect to manipulation, dynamic dexterity was first addressed by Mason and Lynch [90]. Their work was motivated by the following considerations: first, dynamic manipulation is either the most efficient or even the only method to perform certain tasks. Second, dynamic dexterity helps to reduce the complexity of the end effector and/or manipulator. Third, exploiting the task dynamics by following a graspless (non-prehensile) approach is beneficial for the handling of bulky objects. The new manipulation paradigm was demonstrated for tasks such as planar throwing or snatching using minimalistic robots and open-loop control [82, 84]. Dynamical dexterous manipulation skills for different robots were also discussed by Koditschek et al. [71]. In addition to motion planning, they addressed closed-loop control of these tasks based on visual feedback [106].

Huang investigated impulsive manipulation as an example of dynamic manipulation with intermittent contact [57]. In his work, the action planning was decomposed into an inverse sliding problem, which determined the required velocities to drive an object to the desired state, and an impact problem, which determined the parameters of the collision between object and actuator. The presented case studies were restricted to planar problems.

Robotic throwing tasks were investigated by several researchers: Kato and co-workers planned throwing motions and presented experimental results for a two DOF robot and an object with mass 0.01 kg [67]. Katsumata et al. discussed a throwing task with an underactuated two DOF robotic manipulator [68]. Considering more complex robotic structures, Lombai and Szederkenyi followed an offline approach to optimize throwing trajectories for a six DOF robot [79]. Senoo used a four DOF manipulator and a robotic hand to generate throwing trajectories for a tennis ball [120]. They proposed a rolling motion of the ball on the hand in order to increase the release velocity.

The task of robotic ball catching has also been studied by various researchers: Andersson realized two-dimensional catching of ping-pong balls with a robotic arm [3]. Hove and Slotine presented spatial catching with a four DOF manipulator [55]. Burrige et al. described a mirror law for catching a ball with a three DOF robot and a planar paddle end effector [24, 23]. Frese et al. presented ball catching with a seven DOF robot [38]. In their setup, a basket served as end effector to catch the ball. Bäuml et al. discussed kinematically optimal catching with a seven DOF robot and a robotic hand [6]. Riley and Atkeson investigated ball catching with a humanoid robot using a baseball glove to catch [103]. Frank et al. considered shooting and catching as a new technology within logistic chains [37]. In a first experimental study, a Gantry type robot was used to catch objects with up to 0.06 kg.

The large majority of the aforementioned work is based on strategies that applied a form- or force-closure grasp to catch/throw the object. Additionally, the mass of the manipulated object was sufficiently small and thus neglected. In contrast to these approaches, this chapter considers the skill of *nonprehensile* catching. Such a skill is beneficial as it allows to catch objects that are too large for conventional robotic hands. Furthermore, the chapter discusses catching and throwing of objects with non-negligible mass properties.

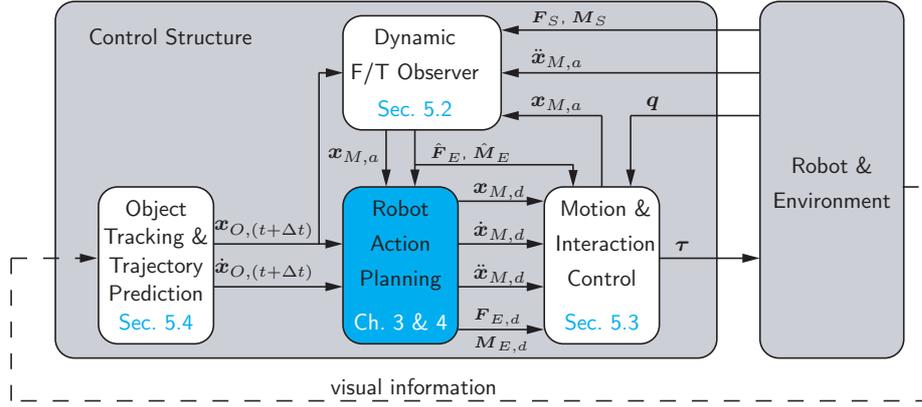


Fig. 3.1: Overall control structure highlighting the action planning module which is addressed in this chapter.

3.2 Trajectory Planning

Fig. 3.1 depicts the overall control structure and highlights the robot action planning module. Here, the term *action planning* refers to the planning of desired motion ($\mathbf{x}_{M,d}$, $\dot{\mathbf{x}}_{M,d}$, $\ddot{\mathbf{x}}_{M,d}$) and/or force ($\mathbf{F}_{E,d}$, $\mathbf{M}_{E,d}$) trajectories. The planning process is based on the modeling foundations discussed in Ch. 2. For dynamic manipulation tasks, the action planning has to be performed online based on the provided sensor feedback. The inputs are the actual end effector pose ($\mathbf{x}_{M,a}$), the predicted object trajectory ($\mathbf{x}_{O,(t+\Delta t)}$, $\dot{\mathbf{x}}_{O,(t+\Delta t)}$), and the estimated forces/torques exchanged with the environment ($\hat{\mathbf{F}}_E$, $\hat{\mathbf{M}}_E$). Details on the environment perception are given in Ch. 5.

The overall goal is to find motions that fulfill the desired optimization criteria and that are dynamically feasible. In this context, dynamically feasible means that the resulting hybrid state trajectory can be realized with a control input \mathbf{u} and the corresponding torques $\boldsymbol{\tau}$ do not exceed hardware limitations [130]. The following section presents the constraints that need to be considered in the planning stage. It also discusses selection criteria that can be used to optimize the trajectory of the hybrid system.

3.2.1 Constraints

Two types of constraints are commonly distinguished: two-sided equality constraints and one-sided inequality constraints, compare Sec. 2.3. Equality constraints are given in the form

$$\mathbf{c}_e(\boldsymbol{\gamma}) = \mathbf{0}, \quad (3.1)$$

where $\boldsymbol{\gamma}$ denotes a set of parameters that define the trajectory. Typically, equality constraints are used to specify positions, velocities, and accelerations at certain points in time. Inequality constraints are given by

$$\mathbf{c}_i(\boldsymbol{\gamma}) \leq \mathbf{0}. \quad (3.2)$$

With these constraints, hardware limitations such as maximum manipulator acceleration or joint limits are considered.

3.2.2 Optimization Criteria

The following paragraphs outline selection criteria w_i for the optimal trajectory planning. Based on these selection criteria, task-specific cost functions

$$J(\boldsymbol{\gamma}, w_1, \dots, w_k) \quad (3.3)$$

are defined. The optimization problem is given by

$$\begin{aligned} \min_{\boldsymbol{\gamma} \in \Gamma} \quad & J(\boldsymbol{\gamma}, w_1, \dots, w_k) \\ \text{with } \Gamma = \quad & \{\boldsymbol{\gamma} \in \mathbb{R}^n, \mathbf{c}_i(\boldsymbol{\gamma}) \leq \mathbf{0}, c_e(\boldsymbol{\gamma}) = 0\} \\ \boldsymbol{\gamma} = \quad & [a_1, \dots, a_m]. \end{aligned} \quad (3.4)$$

The parameter set $\boldsymbol{\gamma}$ defines the trajectory and thus also the corresponding joint angles \mathbf{q} . Consequently, the selection criteria $w_i(\cdot)$ can be formulated either as function of $\boldsymbol{\gamma}$ or \mathbf{q} .

Distance from joint limits. The distance from mechanical joint limits imposes constraints on the trajectory planning. It can also be utilized to ensure that the manipulator stays within preferred regions of its workspace. The measure is given by

$$w_j(\mathbf{q}) = \prod_{i=1}^n w_{j,i}(q_i, q_{i,min}, q_{i,max}) \quad (3.5)$$

where q_i is the angle of the i -th joint and $w_{j,i}$ corresponds to the normalized joint limit measure for the i -th joint.

Dynamic manipulability measure. A global measure for the manipulation ability has been proposed by Yoshikawa in [147]. The measure quantifies the ability for arbitrarily changing position and orientation of the end effector in a given posture. The main drawback of this concept is the fact that it is a kinematic measure ignoring the arm dynamics. Hence, it is not suitable for planning dynamic motions. This shortcoming is addressed by the dynamic manipulability measure w_d introduced in [148]. The scalar measure is defined as

$$w_d(\mathbf{q}) = \frac{1}{w_{d,max}} \sqrt{\det \left(\mathbf{J}(\mathbf{q}) (\mathbf{B}^T(\mathbf{q}) \mathbf{B}(\mathbf{q}))^{-1} \mathbf{J}^T(\mathbf{q}) \right)} \quad (3.6)$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ is the manipulator Jacobian and $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator. The scaling factor $w_{d,max}$ is used to normalize the measure. Fig. 3.2 exemplarily shows the dynamic manipulability measure of the six DOF industrial robot used in this work for different yz -planes and a constant end effector orientation. The reference coordinate system is located in the robot base with the z -direction pointing vertically upwards and the y -direction to the right.

Object orientation. In accordance with the distance from mechanical joint limits, the object orientation can be regarded as both a constraint and an optimization criteria. For a

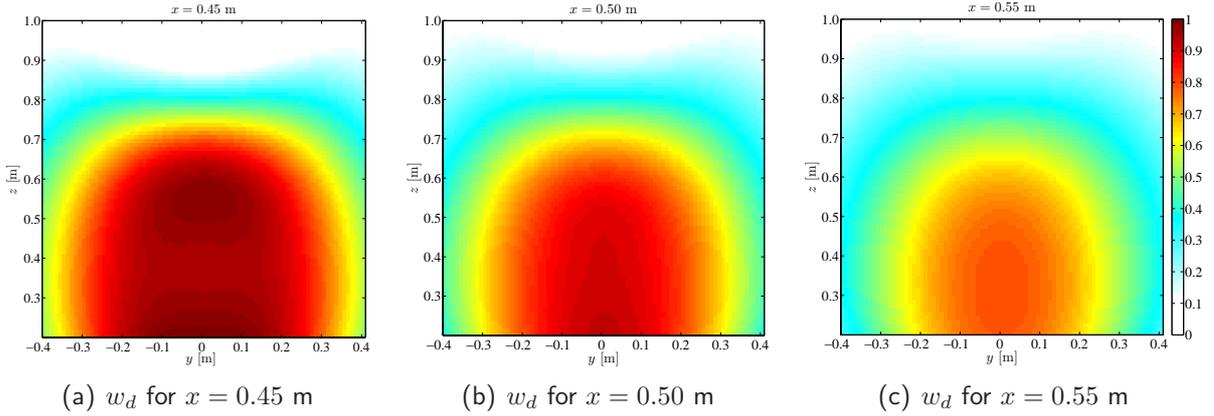


Fig. 3.2: Normalized dynamic manipulability measure w_d for different yz -planes: (a) $x = 0.45$ m, (b) $x = 0.50$ m, (c) $x = 0.55$ m.

specific object, there might exist a range of admissible contact points. Preferences within this range can be considered with a normalized selection criteria $w_o(\mathbf{q}, \mathbf{o}_O)$, where \mathbf{o}_O denotes the orientation of the object.

Energy consumption. Finally, the energy consumption of the system is another criteria for the evaluation of planned trajectories. It can be approximated by

$$w_u(\gamma) = \frac{1}{w_{u,max}} \int_{t_s}^{t_e} \mathbf{u}(\gamma, t)^T \mathbf{u}(\gamma, t) dt, \quad (3.7)$$

where $w_{u,max}$ normalizes the measure and $t_e - t_s$ denotes the duration of the task execution.

3.2.3 Optimization Method

In order to generate near-optimal trajectories in limited time, the optimization method combines offline and online decisions. The method considers tasks that include a free-flight phase. This is a common feature for many dynamic manipulation tasks, the two best-known examples are catching and throwing.

In a first step, equally spaced grid points are defined in the robot workspace, compare Fig. 3.3. For each grid point, the values of the optimization criteria w_j and w_d are determined by averaging the measures for different end effector orientations at that point. For the 14 DOF manipulator, the redundancy is resolved so that distinct values can be assigned to each grid point. These values are stored in look-up tables.

In the second step, motion trajectories are replanned. Here, the initial state of the manipulator for a given task is assumed to be fixed. The trajectories are generated using fifth-order polynomials of which coefficients are determined by position, velocity, and acceleration constraints at initial and final time. The resulting trajectories are checked for dynamic feasibility and their costs are determined based on a combination of w_u , w_o , w_d , and w_j . Again, each grid point is assigned a distinct value by averaging the costs for different final states and the costs are stored in a look-up table.

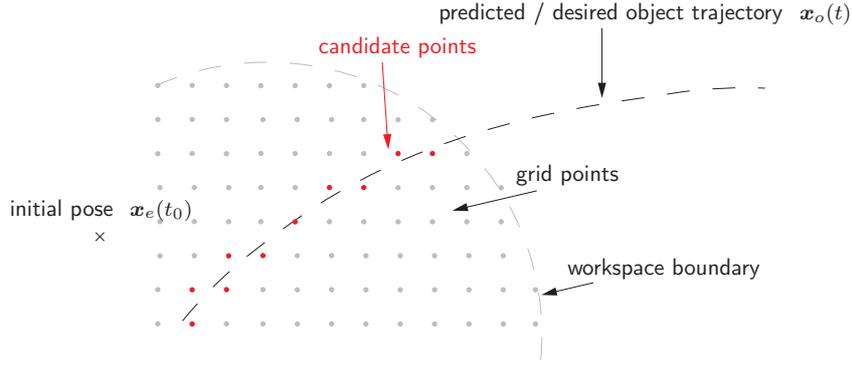


Fig. 3.3: Illustration of the optimization approach.

The third step of the optimization is performed online: the module determines where the desired or predicted free-flight trajectory of the object intersects the workspace of the robot, compare Fig. 3.3. For this part of the free-flight trajectory, the nearest neighbors in the offline computed look-up tables are determined and the ten points with the lowest costs are considered as *candidate points*. For each of these points, the module generates trajectories using fifth-order polynomials. Based on the desired/predicted free-flight trajectory of the object, the coefficients of the polynomials are determined by the task-specific position, velocity, and acceleration constraints at initial and final time. The resulting trajectories of the candidate points are checked for workspace and acceleration constraint violations and the feasible solution with the lowest costs is selected.

3.3 Application for Rigid End Effectors

This section discusses three dynamic manipulation tasks for rigid end effectors: rolling manipulation, throwing, and catching. Here, the following assumptions are made:

- A1** The ball is a rigid body and its geometric center coincides with the center of mass.
- A2** Impacts between ball and manipulator are instantaneous inelastic collisions described by the coefficient of restitution c_r .
- A3** Air resistance is negligible.

The state of the system is given by

$$\mathbf{x} = \left[\mathbf{x}_M^T \quad \mathbf{x}_B^T \quad \dot{\mathbf{x}}_M^T \quad \dot{\mathbf{x}}_B^T \right]^T \in \mathbb{R}^n \quad (3.8)$$

where \mathbf{x}_M is the pose of the end effector(s), \mathbf{x}_B the pose of the ball, and $\dot{\mathbf{x}}_M$ ($\dot{\mathbf{x}}_B$) the corresponding velocities. The overall dimension n of the state is task-dependent. Experimental results for the manipulation tasks are given in Chapter 6.

3.3.1 Balancing

For dynamic manipulation, especially when following a nonprehensile approach, relative motions between object and end effector are an important element, compare Sec. 2.4. With

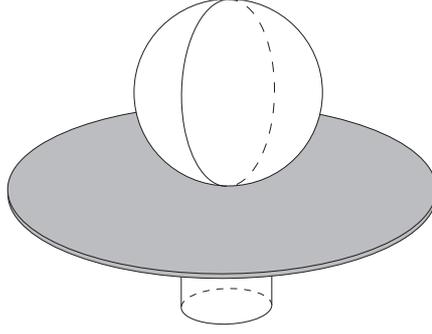


Fig. 3.4: Schematic of the ball-on-plate system.

respect to the applications discussed in this work, rolling manipulation is of particular interest: the stabilization of a spherical object on a planar surface is a preliminary skill for the nonprehensile catching task. To this end, the ball-on-plate system is briefly reviewed. A schematic of the system is shown in Fig. 3.4.

The contact kinematics are derived based on (2.16). The surface parameterization of the ball with radius r_B and the plate are given by

$$\mathbf{c}(u_B, v_B) = \begin{bmatrix} r_B \cos u_B \cos v_B \\ r_B \cos u_B \sin v_B \\ r_B \sin u_B \end{bmatrix} \quad \mathbf{c}(u_M, v_M) = \begin{bmatrix} u_M \\ v_M \\ 0 \end{bmatrix} \quad (3.9)$$

with $-\pi/2 < u_B < \pi/2$ and $-\pi < v_B < \pi$. The geometric parameters of the sphere and the manipulator plate are given by [92]:

$$\begin{aligned} \mathbf{M}_B &= \begin{bmatrix} r_B & 0 \\ 0 & r_B \cos u_B \end{bmatrix}, & \mathbf{K}_B &= \begin{bmatrix} 1/r_B & 0 \\ 0 & 1/r_B \end{bmatrix}, & \mathbf{T}_B &= [0 \quad -1/r_B \tan u_B], \\ \mathbf{M}_M &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & \mathbf{K}_M &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, & \mathbf{T}_M &= [0 \quad 0]. \end{aligned} \quad (3.10)$$

For pure rolling, the condition $v_x = v_y = v_z = \omega_z = 0$ has to be fulfilled (compare Subsec. 2.4.2) and the contact kinematics simplify to:

$$\begin{bmatrix} \dot{u}_B \\ \dot{v}_B \\ \dot{u}_M \\ \dot{v}_M \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/\cos u_B \\ -r_B \sin \psi \\ -r_B \cos \psi \\ -\tan u_B \end{bmatrix} \omega_x + \begin{bmatrix} -1 \\ 0 \\ -r_B \cos \psi \\ r_B \sin \psi \\ 0 \end{bmatrix} \omega_y. \quad (3.11)$$

The design of a balancing control is presented by Lee and Bätz in [158]. The controller utilizes feedback from a wrist force/torque sensor to determine the contact point and stabilizes the ball at the center of the plate. Similar studies of the system dynamics and the design of a stabilizing controller can be found in [98, 72, 87, 12].

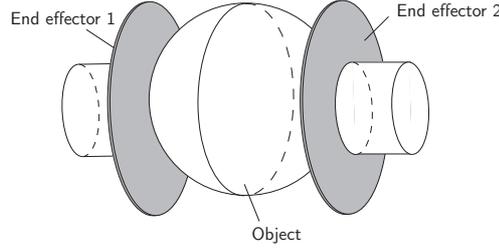


Fig. 3.5: Schematic of dual-handed throwing with a force closure grasp.

3.3.2 Throwing

The two-handed throwing of spherical objects with a basic force-closure grasp is considered. Hence, the object's orientation and its rotational velocity are neglected. The state of the system is then given by

$$\mathbf{x} = \left[\mathbf{x}_{M,1}^T \quad \mathbf{x}_{M,2}^T \quad \mathbf{p}_B^T \quad \dot{\mathbf{x}}_{M,1}^T \quad \dot{\mathbf{x}}_{M,2}^T \quad \mathbf{v}_B^T \right]^T \in \mathbb{R}^{32} \quad (3.12)$$

with $\mathbf{x}_{M,i} = [\mathbf{p}_{M,i}^T \quad \mathbf{o}_{M,i}^T]^T \in \mathbb{R}^7$ and $\dot{\mathbf{x}}_{M,i} = [\mathbf{v}_{M,i}^T \quad \boldsymbol{\omega}_{M,i}^T]^T \in \mathbb{R}^6$. A schematic of the of the dual-handed throwing task is depicted in Fig. 3.5. As first subtask, a grasping motion is executed if a stationary ball position within the robots workspace is tracked. Grasping is realized by *force closure*, applying a desired normal force on the contact surface. The imposed constraint for the normal force F_n is

$$\|m_B(\mathbf{a}_{B,max} + \mathbf{g})\| \leq F_r = \mu_s F_n, \quad (3.13)$$

where μ_s is the static friction coefficient. For the two manipulators, the throwing trajectories in the directions tangential to the contact surfaces are planned based on the optimization method presented in Subsec. 3.2.3. For the direction normal to the contact surfaces, the desired grasping force is specified.

3.3.3 Nonprehensile Catching

This subsection details the one-handed catching of spherical objects. Due to the spherical shape, the object's orientation and its rotational velocity are not considered in the trajec-

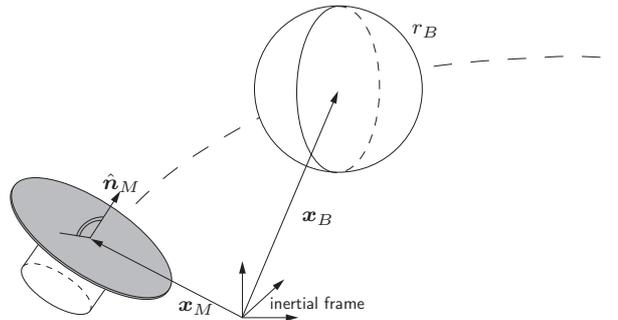


Fig. 3.6: Schematic of the one-handed nonprehensile catching.

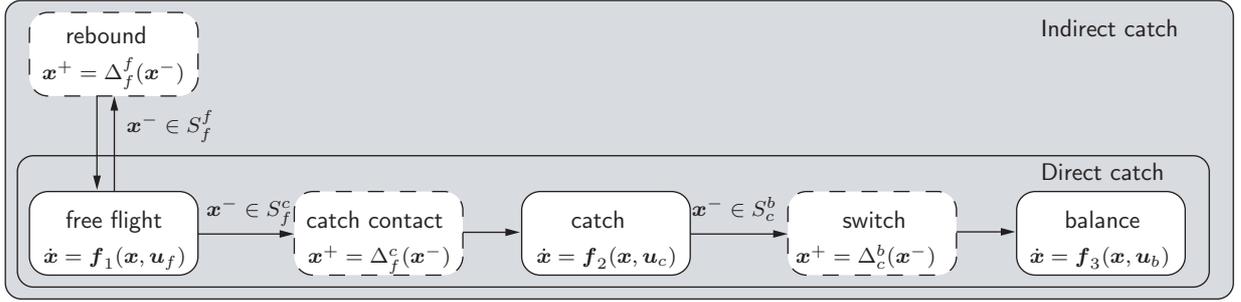


Fig. 3.7: Direct and indirect catch - evolution chart. Solid boxes depict discrete states, dashed boxes depict instantaneous transitions.

tory optimization. They are, however, used for the balancing control and hence included in the state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_M^T & \mathbf{x}_B^T & \dot{\mathbf{x}}_M^T & \dot{\mathbf{x}}_B^T \end{bmatrix}^T \in \mathbb{R}^{26} \quad (3.14)$$

with $\mathbf{x}_M = [\mathbf{p}_M^T \ \mathbf{o}_M^T]^T \in \mathbb{R}^7$ and \mathbf{x}_B analogous. Two different catching strategies are considered: *direct catch* and *indirect catch*. The former strategy catches the ball at first contact, whereas the latter strategy includes an initial rebound and catches the ball at the second contact. The trajectory generation is based on the optimization method described in Subsec. 3.2.3 and the task-specific constraints are detailed in the following two paragraphs.

Direct catch. The evolution chart for a direct catch is depicted in Fig. 3.7. Here, the end effector trajectory at time t_c of the initial contact has to fulfill the following constraints:

$$\begin{aligned} \mathbf{p}_M(t_c) &= \mathbf{p}_B(t_c) + r_B \hat{\mathbf{v}}_B(t_c), & \mathbf{v}_M(t_c) &= \mathbf{v}_B(t_c), \\ \mathbf{a}_M(t_c) &= \mathbf{a}_B(t_c), & \hat{\mathbf{n}}_M(t_c) &\parallel -\mathbf{v}_B(t_c). \end{aligned} \quad (3.15)$$

Here, r_B denotes the radius of the ball, $\hat{\mathbf{v}}_B(t_c) = \mathbf{v}_B(t_c)/\|\mathbf{v}_B(t_c)\|$ is the normalized ball velocity, and $\hat{\mathbf{n}}_M(t_c)$ is the unit normal vector on the end effector plate. Once contact between ball and plate is established, the end effector is decelerated with $a_{cb} = \text{const.}$ in the direction of $\mathbf{v}_M(t_c)$. Hence, the time t_b when the translational velocity of the end effector reaches zero is given by

$$t_b = t_c + \|\mathbf{v}_B(t_c)\|/|a_{cb}|. \quad (3.16)$$

The end effector constraints at time t_b for a direct catch are

$$\begin{aligned} \mathbf{p}_M(t_b) &= \mathbf{p}_M(t_c) + \mathbf{v}_M(t_c)(t_b - t_c) + 0.5\mathbf{a}_M(t_b - t_c)^2, & \mathbf{v}_M(t_b) &= \mathbf{0}, \\ \mathbf{a}_M(t_b) &= \mathbf{0}, & \hat{\mathbf{n}}_M(t_b) &\parallel -\mathbf{g} \end{aligned} \quad (3.17)$$

For $t > t_b$, a balancing controller is applied to stabilize the ball on the plate, for details see Subsec. 3.3.1 and Lee and Bätz [158].

Switching surface	Reset map
$S_f^f = \ \mathbf{v}_M\ + \lambda \ \mathbf{p}_M - \mathbf{p}_B + r_B \hat{\mathbf{n}}_M\ $	$\mathbf{x}^+ = \Delta_f^f(\mathbf{x}^-) =$ $= \begin{bmatrix} \mathbf{x}_M^T & \mathbf{x}_B^T & \dot{\mathbf{x}}_M^T & \mathbf{0}_{1 \times 2} & c_r \ \mathbf{v}_B\ & \boldsymbol{\omega}_B^T \end{bmatrix}^T$
$S_f^c = \ \mathbf{v}_M - \mathbf{v}_B\ + \lambda \ \mathbf{p}_M - \mathbf{p}_B - r_B \hat{\mathbf{v}}_B\ $	$\mathbf{x}^+ = \Delta_f^c(\mathbf{x}^-) = \mathbf{x}^-$
$S_c^b = \ \mathbf{v}_M\ + (x_d - 2)$	$\mathbf{x}^+ = \Delta_c^b(\mathbf{x}^-) = \mathbf{x}^-$

Tab. 3.1: Indirect catch - switching surfaces and reset maps.

Indirect catch. If the direct catch algorithm does not obtain a feasible solution, the module checks whether an indirect catch is possible in order to extend the range of admissible object states. For an indirect catch, the following constraints are set for the end effector trajectory at the initial contact time t_r :

$$\mathbf{p}_M(t_r) = \mathbf{p}_B(t_r) - r_B \hat{\mathbf{n}}_M(t_r), \quad \mathbf{v}_M(t_r) = \mathbf{0}, \quad \mathbf{a}_M(t_r) = \mathbf{0}, \quad \hat{\mathbf{n}}_M(t_r) = -\mathbf{s} / \|\mathbf{s}\|, \quad (3.18)$$

where $\mathbf{s} = \hat{\mathbf{v}}_B(t_r) + \hat{\mathbf{g}}$. With the assumptions A1 and A2, the last constraint ensures that the ball rebounds in vertical z -direction. For the indirect catch, there is an additional phase where the ball is again in free-flight, compare Fig. 3.7. The time t_c of the second contact at $\mathbf{p}_M(t_c) = \mathbf{p}_M(t_r)$ is calculated based on the ball velocity $\mathbf{v}_B(t_r)$ and the coefficient of restitution c_r . The constraints at t_c are then identical to the ones of a direct catch with sole vertical velocity. The switching surfaces and reset maps for the indirect catch are summarized in Tab. 3.1.

3.4 Summary

This chapter discussed non-periodic dynamic manipulation tasks using a generic end effector design. First, an overview of related work was given. Then, the planning of optimal trajectories was discussed. The approach used in this work is based on a combination of online and offline decisions: selection criteria are evaluated offline and look-up tables are created. These look-up tables allow to evaluate the selection criteria online and to generate the desired trajectory. Section 3.3 presented three applications for rigid end effector designs: Subsection 3.3.1 reviewed the ball-on-plate system since the stabilization of a spherical object on a planar surface is a preliminary skill for nonprehensile manipulation. Subsection 3.3.2 addressed the planning of dual-handed throwing motions. Here, the grasping of the object is realized with a basic force closure. Finally, Subsection 3.3.3 presented a novel method for robotic catching: a *nonprehensile* approach was discussed which allows the catching of bulky objects and the trajectory planning for catching spherical objects was outlined. Two catching strategies, direct and indirect, were proposed to increase the range of admissible initial object states. Both approaches considered the nonprehensile catching of spherical objects. Here, the object orientation does not affect the geometry of the contact surfaces. For arbitrarily shaped objects, the object orientation along the trajectory has to be included as additional selection criteria. An experimental evaluation of the studied manipulation tasks is presented in Chapter 6.

4 Periodic Manipulation Tasks with Intermittent Contact

Summary This chapter considers periodic manipulation tasks with an intermittent contact between actuator and object. Ball dribbling is introduced as a novel case study for dynamic object manipulation. The system dynamics and optimal trajectory planning are discussed for both a rigid and a compliant end effector design. The virtual holonomic constraints approach is used to design a stabilizing controller for the dribbling task. In addition, the classic juggling task is considered: the chapter reviews control strategies proposed in the literature and presents a new approach for optimal trajectory planning based on a non-local stability analysis.

The control of rhythmic tasks has been a very active research area over the last two decades. A large amount of these research activities focused on legged robot locomotion. However, besides robotic walking, there are also numerous works that investigated rhythmic manipulation tasks which are commonly summarized with the term juggling. Bühler coined the term for tasks that require interaction with an object or multiple objects that would otherwise fall freely in the earth's gravitational field [21]. For such tasks, the continuous motion of an actuator is used to control the continuous motion of one or multiple objects through a temporary contact. Due to this intermittent or impulsive contact, the systems possess hybrid dynamics described by discrete states with continuous dynamics and instantaneous transitions.

The interest in juggling tasks originates from different sources: the first is the fact that they provide an excellent scenario to investigate the role of feedback in rhythmic tasks. Since the object is only controlled during a fraction of the cycle, the question arises, in which phases sensor feedback is needed to stabilize a desired periodic motion. A second, closely related aspect is the question of controllability of these systems. Due to the underactuation of the system and the impulsive or intermittent character of the contact phases, the control of these tasks is challenging. In addition, studying these manipulation tasks can provide clues for the mechanical design of the manipulator. Finally, such periodic tasks are helpful to increase the understanding of human movement control. The task of juggling a ball, for instance, poses all the fundamental questions that are raised in the study of human motion coordination, see [117]. Finally, due to their periodicity, these tasks are well suited for the evaluation of learning algorithms.

The chapter is organized as follows: Section 4.1 provides an overview on related work.

In Section 4.2, the planning of optimal trajectories for periodic manipulation tasks is addressed. Section 4.3 discusses the stability of periodic solutions of ordinary differential equations. Next, Section 4.4 addresses the stability of periodic solutions of hybrid dynamical systems. In addition to local stability, the section also considers a region of attraction analysis. Section 4.5 then applies these methods to tasks with a rigid manipulator. Besides the classic juggling, the task of dribbling a ball is considered as it introduces additional challenges. The distinctive feature of the dribbling task is the existence of a second, autonomous impact between the object and the environment. Finally, Section 4.6 details the realization of the dribbling task with a compliant end effector.

4.1 Related Work

Holmes was the first to investigate the dynamics of repeated impacts on a sinusoidally vibrating table [54]. With this work, he introduced the bouncing ball model as a case study in nonlinear dynamics which has since become very popular. Holmes proved that, for sufficiently large excitation velocities and a coefficient of restitution close to one, the system possesses both periodic solutions as well as irregular non-periodic and chaotic motions. Inspired by Holmes pioneering work, various researchers have considered this or similar tasks: Bapat and coworkers followed up the work of Holmes, focusing on the sinusoidal motions which led to stable periodic collisions [5]. Their results already implied that such a sinusoidal actuation requires negative actuator acceleration at impact to obtain open-loop stability. Based on a refinement of Holmes original model, the numerical simulations of Luo and Han led to an extended range of stable periodic motions [80].

Bühler et al. were the first to investigate the stabilization of robotic juggling tasks [20, 21, 22]. In their work, they considered the trajectory control of a puck which was constrained to lie on a plane turned into the earth's gravitational field. The manipulator was a one DOF robot with a revolute joint. They proposed the so called mirror law algorithm for stabilizing the desired periodic motions of the puck. Here, the idea is that the manipulator describes a distorted reflection (mirroring) of the objects trajectory. The control algorithm was verified for a planar experimental setup. However, it should be noted that the mirror law does not provide open-loop stability and hence requires permanent tracking of the ball. Following the work of Bühler, Rizzi et al. extended the mirror law to accomplish spatial juggling of two balls with a three DOF robot manipulator [104, 105, 107, 23].

Atkeson and Schaal investigated the learning of different periodic manipulation tasks, e.g. devil sticking and classic juggling. Here, they demonstrated that task-level learning improves the performance of a juggling robot [2, 116, 115]. In addition, Schaal and Atkeson discussed open-loop stability for the classic juggling task. Generalizing the results of Holmes and Bapat, they showed that, for a line juggler, an arbitrary periodic motion of the manipulator results in open loop stability if the manipulator acceleration at impact is within a certain negative range [114]. Considering systems with one DOF object motion, this is an fundamental advantage compared to the mirror algorithm as this approach allows sensorless stabilization of a desired periodic orbit. For planar or spatial juggling systems, however, it is important to note that the open-loop control can not stabilize a particular

contact point in the impact plane. Horizontal deviations still have to be compensated by an additional control or an appropriate mechanical design, see e.g. [102]. Schaal and Atkeson also provided a non-local stability analysis for the open loop stable control strategies [117]. However, this analysis is of limited use since it does not determine a region of attraction but only allows relative comparisons of different controller designs. Based on these results, Schaal and Sternard also published a number of studies that focused on human motion coordination in a classic juggling task [117, 133, 134].

Driven by the interest in minimalist manipulation systems, Lynch and Black discussed the planar juggling system with one actuated revolute joint introduced by Bühler. Here, the planar juggler served as a case study for systems that are controllable, but insufficiently actuated to be locally controllable. Applying the idea of forced recurrence, they showed controllability of the system on a large subspace of the state space and developed a stabilizing controller [81]. Brogliato et al. also discussed controllability and proposed stabilizing feedback laws for the one DOF and the planar juggler [19, 150].

A special type of juggling system, the so called wedge billiard, was first introduced by Lehtitiet and Miller and later studied in detail by Sepulchre et al. [121, 77]. Such dynamical billiards are part of the mathematical theory of particle trajectories within a closed reflective boundary. Ronsse and Sepulchre studied different types of actuation and investigated the importance of timing the control feedback [108, 109, 110]. They also discussed feedback control strategies for the bouncing ball system and discussed the influence of the impact acceleration on the robustness of the system to parameter uncertainties [112].

An example of robot dribbling was presented by Shiokata et al. for experimental evaluation of a high-speed vision system [122]. For the experiment, a ping-pong ball and an actuator with four DOF were used. The task served as a benchmark study for the high-speed vision system and did not discuss the trajectory planning or the design of a stabilizing controller.

Besides their periodic nature, the works discussed so far all have the characteristic property of an intermittent and/or impulsive contact in common. For the sake of completeness, it has to be mentioned that there exist also numerous periodic/rhythmic dynamic manipulation tasks with continuous contact phases. An interesting example is the work of Zacksenhouse et al. who investigated robotic yo-yo playing [62, 63, 61]. However, these works are not further discussed as the focus of the chapter is on periodic manipulation tasks with intermittent contact.

4.2 Trajectory Planning

The trajectory planning for periodic tasks is similar to the one for non-periodic which was presented in Sec. 3.2. Again, the planning is based on the modeling foundations discussed in Ch. 2. As detailed in Sec. 3.2, equality constraints $\mathbf{c}_e(\boldsymbol{\gamma}) = \mathbf{0}$ are used to specify positions, velocities, and accelerations at certain points in time and inequality constraints $\mathbf{c}_i(\boldsymbol{\gamma}) \leq \mathbf{0}$ are used to incorporate hardware limitations. The same approach and the same selection criteria as for the tasks studied in Sec. 3 can be applied for the trajectory optimization of periodic manipulation tasks, compare (3.4)-(3.7). The trajectory planning for periodic manipulation tasks also includes on- and offline decisions. However, the periodic nature of

the tasks allows more offline decisions in the planning process.

Considering the open-loop control of a periodic manipulation task, the manipulator motion is described by an arbitrary but periodic trajectory. Hence, the motion of the manipulator can be described by a finite Fourier series

$$x_M(t) = \sum_{k=1}^L A_k \sin(k\omega t + \phi_k). \quad (4.1)$$

Consequently, position, velocity, and acceleration of the manipulator at time t_n (at the n -th impact) are given by

$$\begin{aligned} x_{M[n]} &= \sum_{k=1}^L A_k \sin(k\omega t_n + \phi_k) = f(\omega t_n) = f(\theta_{[n]}), \\ \dot{x}_{M[n]} &= \omega \sum_{k=1}^L A_k k \cos(k\omega t_n + \phi_k) = \omega f'(\omega t_n) = \omega f'(\theta_{[n]}), \\ \ddot{x}_{M[n]} &= \omega^2 \sum_{k=1}^L -A_k k^2 \sin(k\omega t_n + \phi_k) = \omega^2 f''(\omega t_n) = \omega^2 f''(\theta_{[n]}), \end{aligned} \quad (4.2)$$

where the expressions $f'(\theta_{[n]}) = \frac{\partial f(\theta_{[n]})}{\partial \theta_{[n]}}$ and $f''(\theta_{[n]}) = \frac{\partial^2 f(\theta_{[n]})}{\partial^2 \theta_{[n]}}$ have been used [134].

4.3 Stability of Periodic Solutions of Ordinary Differential Equations

This section outlines definitions that are preliminary for the stability analysis of hybrid systems in Sec. 4.4. A detailed discussion of these definitions can be found in the books by Parker and Chuo, Khalil, and Slotine [99, 69, 129]. In the following, an autonomous continuous-time dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ as defined in (2.2) is considered. By assuming a state feedback control $\mathbf{u} = \mathbf{k}(\mathbf{x})$, the explicit dependency of the system on the control input \mathbf{u} is omitted.

Definition 4.1 (Periodic solution, Periodic orbit, Limit cycle) The flow $\phi_t(\mathbf{x}_0)$ is a periodic solution (orbit) Γ of an autonomous system if, for all t ,

$$\phi_t(\mathbf{x}_0) = \phi_{t+T}(\mathbf{x}_0) \quad (4.3)$$

for some period $T > 0$. The restriction $T > 0$ is needed to prevent the classification of an equilibrium point as periodic solution. A periodic solution is *isolated* if it possesses a neighborhood that contains no other periodic solution. In the autonomous case, an isolated periodic solution is called a limit cycle. The limit set corresponding to a limit cycle is the closed curve defined by $\phi_t(\mathbf{x}_0)$ over one period. \square

Typically, in a nonlinear dynamical system, there exist other limit sets in addition to the periodic solution such as equilibrium points, quasi-periodic solutions and chaos. The focus

in this chapter, however, is the study of periodic solutions.

Definition 4.2 (Invariant set) A set M is said to be an invariant set with respect to an autonomous ODE (2.2) if

$$\mathbf{x}(t_0) \in M \Rightarrow \mathbf{x}(t) \in M, \quad \forall t \geq \mathbb{R} \quad (4.4)$$

Hence, a periodic solution $\phi_t(\mathbf{x}_0)$ is a closed invariant set of the dynamic system (2.2). In the following, the periodic solution is denoted by Γ .

Definition 4.3 (LaSalle's invariance principle) Consider an autonomous ODE as defined in (2.2) and let $V(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ be a locally positive definite function. Assume that, on the compact set

$$\Omega_l = \{\mathbf{x} \in \mathbb{R}^n : V(\mathbf{x}) \leq l\}, \quad \text{it holds that } \dot{V}(\mathbf{x}) \leq 0 \quad \forall \mathbf{x} \in \Omega_l.$$

Define $R = \{\mathbf{x} \in \Omega_l : \dot{V}(\mathbf{x}) = 0\}$ and let M be the largest invariant set in R . Then, every solution $\mathbf{x}(t)$ originating in Ω_l tends to M as $t \rightarrow \infty$. \square

Definition 4.4 (Discrete-time dynamical system) A discrete dynamical system is defined by

$$\mathbf{x}_{[k+1]} = \mathbf{f}_d(\mathbf{x}_{[k]}) \quad k = 0, 1, 2, \dots \quad (4.5)$$

where $\mathbf{x}_{[k]} \in \mathbb{R}^n$ is the state and $\mathbf{f}_d : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ maps the current state $\mathbf{x}_{[k]}$ to the next state $\mathbf{x}_{[k+1]}$. \square

Poincaré map. The Poincaré map is a widely used technique for the analysis of dynamical systems. Here, the flow of an n -th order dynamical system is replaced with an $(n - 1)$ -th order discrete-time system which is called the Poincaré map. Its main advantages are the reduction of order and the fact that it bridges the gap between continuous- and discrete-time systems [99].

Definition 4.5 (Poincaré map, First return map for autonomous systems)

Consider an n -th order autonomous system with the limit cycle (periodic orbit) Γ as shown in Fig. 4.1. Let \mathbf{x}^* denote a point on the limit cycle and let Σ be an $(n - 1)$ -dimensional cross section (hyperplane) transversal to Γ at \mathbf{x}^* . A cross section is transversal, if $\mathbf{n}(\mathbf{x})^T \mathbf{f}(\mathbf{x}) \neq 0$, where \mathbf{n} is the normal vector to Σ . Then, the Poincaré map $\mathbf{P} : U \rightarrow \Sigma$ is defined in a neighborhood $U \subset \Sigma$ of \mathbf{x}^* as

$$\mathbf{P}(\mathbf{x}) := \phi_{T(\mathbf{x})}(\mathbf{x}). \quad (4.6)$$

The time $T(\mathbf{x})$ denotes the first time the orbit and the cross section intersect after starting the integration in $\mathbf{x} \in U$. The mapping has a fixed point if $\mathbf{P}(\mathbf{x}^*) = \mathbf{x}^*$. \square

Due to the definition of the Poincaré map, its limit sets correspond to the limit sets of the underlying flow. Furthermore, the asymptotic stability of the fixed point can be shown to be equivalent to the asymptotic orbital stability of the periodic solution. A fixed point

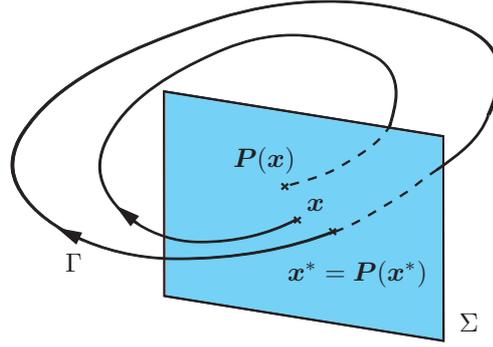


Fig. 4.1: Poincaré map

\mathbf{x}^* of the nonlinear difference equation $\mathbf{x}_{[n+1]} = \mathbf{P}(\mathbf{x}_{[n]})$ can be shown to be asymptotically stable if all eigenvalues of its linearization

$$\left. \frac{\partial \mathbf{P}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*} \quad (4.7)$$

are strictly within the unit circle [99, 130].

Topological equivalence of flows. The concept of equivalence is a useful tool for the comparison of (hybrid) dynamical systems. Two orbits of a smooth or hybrid dynamical system are topologically orbitally equivalent (TOE) if there exists a orientation-preserving homeomorphism that maps one orbit into the other [60, 128]. Employing the TOE helps to simplify the analysis of periodic solutions: if such a homeomorphism exists, the results of a particular periodic solution can be generalized to other periodic orbits, see e.g. Schaal [134].

4.4 Stability of Periodic Solutions of Hybrid Dynamical Systems

The following section outlines methods for local and global stability analysis of hybrid systems. A hybrid system as stated in Definition 2.3 is considered. However, the explicit dependence on the control inputs \mathbf{u} and u_d is omitted by assuming a state-dependent control law in the form $\mathbf{u} = \mathbf{k}(\mathbf{x}, x_d)$ and $u_d = k(\mathbf{x}, x_d)$. The hybrid system is then given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}_i(\mathbf{x}, x_d) && \text{if } s_i^j(\mathbf{x}, x_d) \notin 0 \quad \forall \{i, j\}, \\ \zeta^+ &= \begin{bmatrix} \mathbf{x}^+ \\ x_d^+ \end{bmatrix} = \begin{bmatrix} \Delta_i^j(\mathbf{x}^-, x_d) \\ j \end{bmatrix} && \text{if } s_i^j(\mathbf{x}, x_d) = 0. \end{aligned} \quad (4.8)$$

4.4.1 Local Stability Analysis

The concepts for local stability analysis of periodic solutions of hybrid system are analogous to the ones for ODEs discussed in Sec. 4.3.

Definition 4.6 (Hybrid periodic solution, Hybrid periodic orbit) A solution of ζ is a hybrid periodic solution Γ with period length $T > 0$ if

$$\zeta(t + T) = \zeta(t), \quad (4.9)$$

holds for all $t \in \mathbb{R}$. □

This definition of a hybrid periodic solution implies that there exists a switching sequence of the discrete state x_d which corresponds to the periodic solution.

A general definition for the stability of invariant sets of hybrid dynamical systems was given by Simic et al. [127]. In most works that consider the stability of hybrid periodic solutions, the stability of the hybrid limit cycle is determined with the stability of the discrete mapping of the hybrid Poincaré map [130, 51].

Poincaré map for hybrid systems. The definition of the hybrid Poincaré map is in accordance with the one for smooth dynamical systems stated in Def. (4.5). Assume a cross section Σ that is transversal to the flow of the hybrid system. Then, the hybrid Poincaré map is defined to map initial values on Σ to the next occurrence of a crossing of Σ . A detailed discussion on the construction of the hybrid Poincaré map is provided by Sobotka in [130].

4.4.2 Non-Local Stability Analysis

The Poincaré map is a powerful tool and a widely used technique for the local stability analysis of periodic solutions for hybrid dynamical systems. However, the eigenvalues of the linearized system dynamics provide only a local stability measure. In order to evaluate the robustness of a hybrid periodic orbit with respect to disturbances or to modeling errors, it is important to know how the system behaves in the non-local neighborhood of the periodic solution.

Region of attraction. The behavior in the neighborhood of the hybrid limit cycle can be evaluated with a region of attraction analysis. Such an analysis addresses the shortcomings of the Poincaré map and determines a region in which initial states are *attracted* to the periodic orbit. Following a numerical approach, the region of attraction is estimated by simulating the system's hybrid flow for initial states in the vicinity of the desired fixed point. For a dense grid of sample points, the convergence/divergence of the corresponding flows determines an attracting region for the hybrid limit cycle. A cell mapping method can be used to reduce the computational burden of the numerical approach [56]. Schwab and Wisse employed this method to evaluate the attracting region of a periodic orbit for a passive walking model [118]. The approach can be used for systems with either open- or closed-loop control.

4.5 Application for Rigid End Effectors

In this section, a rigid manipulator design is assumed and two periodic manipulation tasks with intermittent contact are considered: classic juggling and dribbling. For both tasks, the dynamic model and the first return map are derived. Based on the model, a local and global stability is performed. For the model and the stability analysis, the following assumptions are made:

- A1** The ball and the manipulator are modeled as one DOF systems such that only vertical motions are taken into account.
- A2** The ball is a rigid body and its geometric center coincides with the center of mass.
- A3** Impacts between ball and manipulator and (possible) impacts between ball and ground are instantaneous inelastic collisions described by the coefficient of restitution c_r .
- A4** Air resistance and rotational ball velocity are negligible.

4.5.1 Classic Juggling Task

A schematic of the juggling system is depicted in Fig. 4.2. The manipulator is used to control the motion of the object through an instantaneous contact.

Review of control strategies. Different solutions have been reported in literature to realize stable periodic motions for the juggling task. A *sinusoidal* trajectory

$$x_M(t) = A \sin(\omega t + \phi_0) + x_{M,0} \quad (4.10)$$

which does not require any sensory feedback has been first proposed by Holmes and later by Schaal et al. [54, 114]. Next, Bühler et al. presented a *mirror law*

$$x_M(t) = \left(-\frac{1 - c_r}{1 + c_r} - \kappa (E^* - E(t)) \right) x_B(t), \quad (4.11)$$

that requires permanent tracking of the ball [21, 22, 104, 105, 107]. Here, E^* and $E(t)$ denote the desired and the actual ball energy. Finally, a trajectory based on state-feedback

$$x_M(t) = x_{M[n+1]} + \dot{x}_{M[n+1]} (t - t_{[n+1]}) + \frac{1}{2} \ddot{x}_{M[n+1]} (t - t_{[n+1]})^2 \quad (4.12)$$

for $t_{[n]} < t < t_{[n+1]}$ has been discussed by Ronsse et al. [110, 111]. For the three strategies, the manipulator position x_M and velocity \dot{x}_M at impact are determined by the desired periodic motion. The manipulator acceleration \ddot{x}_M , in contrast, is a free parameter. In the following, the optimal open-loop control strategy for the juggling task is evaluated.

Dynamic model. The state of the system is described by the position and velocity of the manipulator and the ball,

$$\mathbf{x} = \left[x_M \quad x_B \quad \dot{x}_M \quad \dot{x}_B \right]^T. \quad (4.13)$$

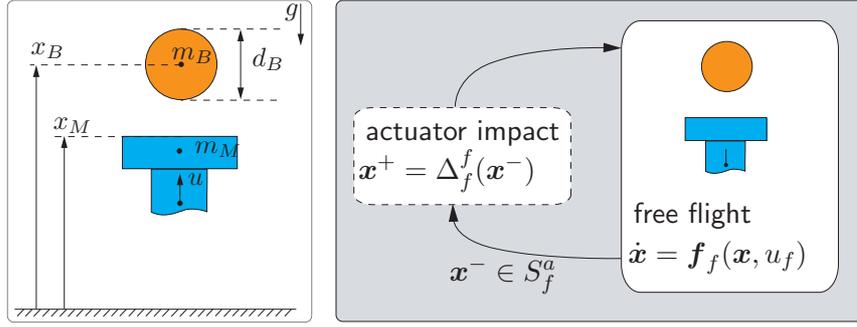


Fig. 4.2: Classic juggling task: Schematic (left) and evolution chart (right) of the juggling system with a rigid actuator. Solid boxes depict discrete states, dashed boxes depict instantaneous transitions.

For the juggling task, a state vector of dimension three is sufficient to describe the system [134]. However, using a non-minimal state vector of dimension four facilitates the comparison with the dribbling task in Subsec. 4.5.2 and 4.5.3. A Poincaré cross section directly before the actuator impact is defined. In the following equations, the superscript $-$ denoting the pre-impact time t^- is omitted to simplify the notation. The state immediately after the impact event is denoted by the superscript $+$. With these definitions, the position of ball and manipulator at the cross section are related via

$$x_{B[n]} = x_{M[n]} + d_B/2 \quad (4.14)$$

and the state dimension is reduced to three. Sternard and Schaal proposed a modification of the state vector to facilitate the stability analysis: the manipulator position $x_{M[n]}$ at the n -th impact is replaced with the angle $\theta_{[n]}$ of the Fourier series at the n -th impact [134]. Considering (4.2), $\theta_{[n]} = \omega t_n$ also specifies the ball position at impact, however in an implicit way. This results in the modified state vector

$$\mathbf{x}(t_n) = \mathbf{x}_{[n]} = \begin{bmatrix} \theta_{[n]} \\ \dot{x}_{M[n]} \\ \dot{x}_{B[n]} \end{bmatrix} \quad (4.15)$$

and discrete relations in the following implicit form $\mathbf{0} = \mathbf{f}(\mathbf{x}_{[n]}, \mathbf{x}_{[n+1]})$, precisely

$$\begin{aligned} \theta_{[n+1]} &= \theta_{[n]} + T\omega \\ \dot{x}_{M[n+1]} &= \omega f'(\theta_{[n+1]}) \\ \dot{x}_{B[n+1]} &= -\sqrt{((1 + c_r)\dot{x}_{M[n]} - c_r\dot{x}_{B[n]})^2 - 2g(x_{M[n+1]} - x_{M[n]})} \end{aligned} \quad (4.16)$$

where $x_{M[n+1]} = f(\theta_{[n+1]})$ and $x_{M[n]} = f(\theta_{[n]})$. The periodic time T is given by

$$T = \frac{1}{g} \left(\dot{x}_{B[n]}^+ + \sqrt{(\dot{x}_{B[n]}^+)^2 - 2g(x_{M[n+1]} - x_{M[n]})} \right) \quad (4.17)$$

with $\dot{x}_{B[n]}^+ = (1 + c_r)\dot{x}_{M[n]} - \dot{x}_{B[n]}$.

Local stability analysis. For the local stability analysis, the system (4.16) is linearized around the fixed point using the implicit function theorem [134]. The linearization is given by

$$\left. \frac{\partial \mathbf{P}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*} = \begin{bmatrix} 1 & -\omega c_r(1+c_r)/g & (1+c_r)^2\omega/g \\ 0 & c_r^2 & -c_r(1+c_r) \\ \omega f''(\theta_{[n]}) & -\omega^2 f''(\theta_{[n]})c_r(1+c_r)/g & \omega^2 f''(\theta_{[n]})(1+c_r)^2/g \end{bmatrix}. \quad (4.18)$$

The system (4.16) has an asymptotically stable fixed point \mathbf{x}^* if all eigenvalues of the linearization (4.18) are strictly within the unit circle, compare Def. 4.5 and [99]. This is the case if the condition

$$\frac{-2(1+c_r^2)g}{(1+c_r)^2} < \ddot{x}_{M[n]} < 0 \quad (4.19)$$

is fulfilled [116]. Depending on c_r , the lower bound varies between $-g$ ($c_r = 1$) and $-2g$ ($c_r = 0$).

Non-local stability analysis. A non-local stability analysis for the classic juggling task was presented by Schaal et al. [117]. The analysis was based on a discrete-time Lyapunov function L_n for the linearized system. The state was sampled around the equilibrium state \mathbf{x}^* and, for each of sample i , the relative change $\Delta L_i = L_{i,[n+1]} - L_{i,[n]}$ was determined. By summarizing over all samples, a stability measure $S = \sum_i \Delta L_i$ is obtained. This approach has several drawbacks: first, the range of validity of the analysis is limited since the Lyapunov function is based on the linearized system. Second, the overall stability measure does not provide information about which initial states converge to the periodic orbit. Finally, the relative change ΔL_i over one period is of limited use since a convergence towards the orbit could occur over multiple cycles. Due to these limitations, the method presented in Subsec. 4.4.2 is better suited to estimate the region of attraction.

Fixed points that possess a large region of attraction are desirable since larger disturbances and modeling errors can be compensated. The following two paragraphs describe the chosen numerical approach to analyze the region of attraction for the juggling task.

Manipulator acceleration. For this analysis, the coefficient of restitution was assumed to be $c_r = 0.7$ and the periodic time T was set to 0.4 s. These assumptions define the desired fixed point as

$$\mathbf{x}^* = [x_M^* \ x_B^* \ \dot{x}_M^* \ \dot{x}_B^*]^T = [0 \ 0 \ 0.3462 \ -1.962]^T. \quad (4.20)$$

The relation derived in (4.19) states a regime of manipulator accelerations \ddot{x}_M^* that result in local stability of the fixed point. In order to investigate the interdependence between the manipulator acceleration and the size of the region of attraction, the manipulator trajectory is optimized for a number of fixed \ddot{x}_M^* . To this end, the system is initialized at equidistant grid points \mathbf{x}_j in the vicinity of the fixed point \mathbf{x}^* and then simulated for $N_c = 20$ cycles. In the vicinity of the desired impact, the periodic manipulator trajectory

(4.2) can be approximated by a sixth-order polynomial [117]

$$x_M(t) = \sum_{k=0}^6 a_k (t - t_n)^k \quad \text{for} \quad ||t - t_n|| \leq t_{thresh}. \quad (4.21)$$

For a sampling point \mathbf{x}_j , the initial and final error are defined as

$$e_{i,j} = \sqrt{(\mathbf{x}_{j[1]} - \mathbf{x}^*)^T (\mathbf{x}_{j[1]} - \mathbf{x}^*)}$$

$$e_{f,j} = \begin{cases} 0, & \text{if } \sqrt{(\mathbf{x}_{j,[N_c]} - \mathbf{x}^*)^T (\mathbf{x}_{j,[N_c]} - \mathbf{x}^*)} \leq e_{thresh} \\ e_{j,max} & \text{else,} \end{cases} \quad (4.22)$$

where $e_{j,max}$ is the maximum error that occurred at the last five impacts. Using the maximum error allows to identify chaotic behavior that might, by accident, show a relative small error for the final cycle of the simulation. With (4.22), the relative change of the error $e_{rel,j} = e_{f,j}/e_{i,j}$ can be determined for each grid point \mathbf{x}_j . The trajectory optimization utilizes a cost function that combines the relative change of the error for each sample point \mathbf{x}_j . The optimization problem is given by

$$\min_{\gamma \in \Gamma} J(\gamma) = \lambda \sum_{j=1}^{N_s} \left(\frac{e_{f,j}}{e_{i,j}} \right)^2$$

$$\text{with } \Gamma = \{ \gamma \in \mathbb{R}^7, \mathbf{c}_i(\gamma) \leq \mathbf{0}, c_e(\gamma) = 0 \}$$

$$\gamma = [a_0, \dots, a_6]$$
(4.23)

Here, $\lambda = 10^2$ and the number of grid points is $N_s = 357$ (21 along $\Delta \dot{x}_B$ and 17 along ΔT). The inequality constraints $\mathbf{c}_i(\gamma)$ restrict the manipulator trajectory to feasible motions. The equality constraints $\mathbf{c}_e(\gamma)$ are determined by the desired fixed point \mathbf{x}^* and the desired manipulator acceleration,

$$\mathbf{c}_i(\gamma) = \begin{bmatrix} ||a_3|| - 10^2 \\ \vdots \\ ||a_6|| - 10^2 \end{bmatrix} \quad \mathbf{c}_e(\gamma) = \begin{bmatrix} a_0 - x_M^* \\ a_1 - \dot{x}_M^* \\ a_2 - \ddot{x}_M^* \end{bmatrix}. \quad (4.24)$$

The results of the optimization for six manipulator accelerations in the range from $\ddot{x}_M = 0.0 \text{ m/s}^2$ to $\ddot{x}_M = -7.5 \text{ m/s}^2$ are illustrated in Fig. 4.3. The coefficient of restitution was assumed to be $c_r = 0.7$ and the periodic time T was set to 0.4 s. For the cases (b) to (f), regions of attraction (white areas) around the fixed point \mathbf{x}^* are found. However, no attracting region is found for case (a) with $\ddot{x}_M = 0 \text{ m/s}^2$. This corresponds to the result of (4.19) which determines zero manipulator acceleration at the impact as locally unstable. While there is a relatively large region of attraction for $\ddot{x}_M = -1.5 \text{ m/s}^2$, it reduces with decreasing \ddot{x}_M . This is also illustrated in Tab. 4.1, where the optimized costs J for different manipulator accelerations \ddot{x}_M are summarized. Based on these results, a manipulator trajectory with an acceleration \ddot{x}_M^* at impact in the range of $[-0.5, -2] \text{ m/s}^2$ is desirable as this maximizes the region of attraction.

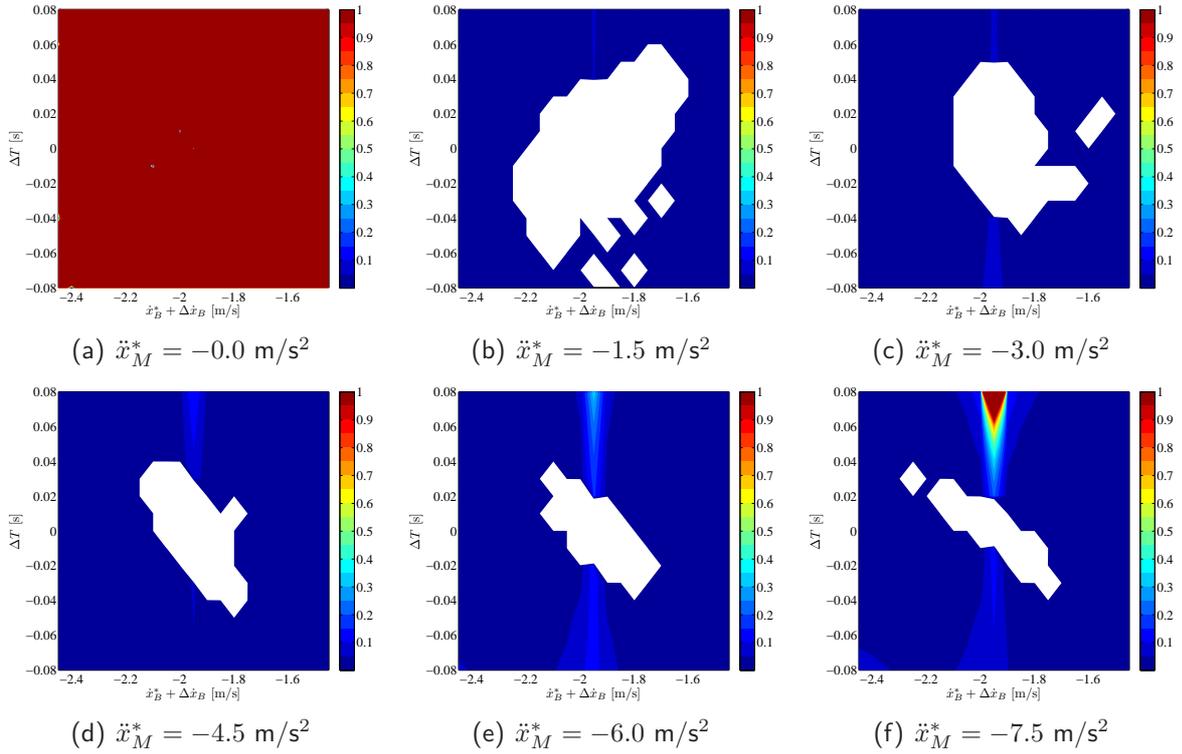


Fig. 4.3: Optimized regions of attraction for the juggling task with a rigid manipulator for different manipulator accelerations \ddot{x}_M . The coefficient of restitution is $c_r = 0.7$ and the periodic time T was set to 0.4 s. The plots illustrate the relative change of the error e_{rel} . For initial states in the white areas, the trajectory converges to the desired orbit. In contrast, dark red areas correspond to an increasing error and thus indicate unstable regions.

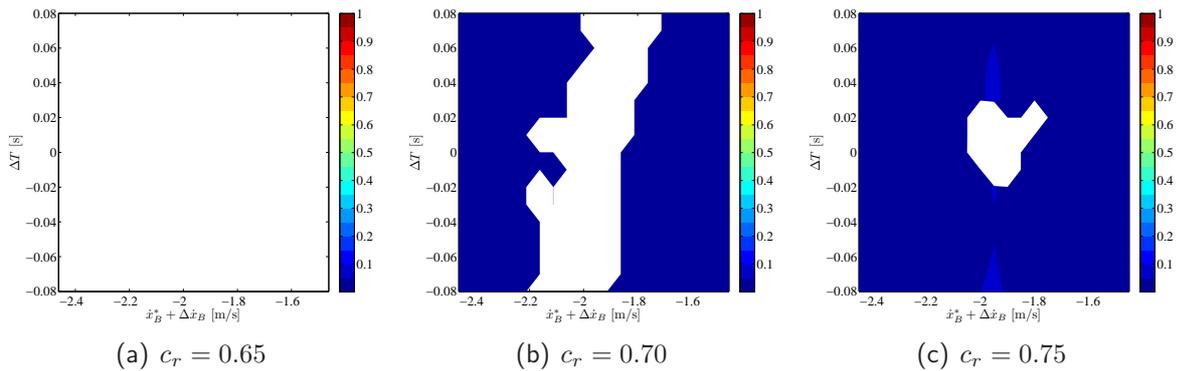


Fig. 4.4: Optimized regions of attraction for the juggling task with a rigid manipulator for different coefficients of restitution c_r . The periodic time T was set to 0.4 s. The plots illustrate the relative change of the error e_{rel} . For each case, the optimized trajectory reduces the initial error. However, the rate of convergence decreases as c_r increases (illustrated by the decreasing size of the white area).

	Manipulator acceleration \ddot{x}_M [m/s ²]						
	0.0	-1.5	-3.0	-4.5	-6.0	-7.5	-9.0
Costs J	$9.4 \cdot 10^7$	3.2	7.3	13.5	69.3	$3.8 \cdot 10^3$	$2.5 \cdot 10^5$

Tab. 4.1: Optimized costs J for different manipulator accelerations \ddot{x}_M with $c_r = 0.7$.

	Coefficient of restitution c_r						
	0.60	0.625	0.65	0.675	0.70	0.725	0.75
Costs J	0	0	0	0.007	0.61	3.09	9.56

Tab. 4.2: Optimized costs J for different coefficients of restitution c_r with optimized \ddot{x}_M^* .

Coefficient of restitution. In contrast to the manipulator acceleration, the coefficient of restitution c_r is a task-specific parameter which is determined by the properties of the colliding bodies, compare Sec. 2.6. Consequently, the value of c_r can not be the subject of an optimization process. However, (4.19) showed that c_r influences the local stability. Hence, it is interesting to see which effect c_r has on the region of attraction. To this end, a second numerical optimization was performed for different values of c_r . In this optimization, the manipulator acceleration \ddot{x}_M is a free parameter. The results of the optimization for three values of c_r are depicted in Fig. 4.4(a)-(c). Again, the periodic time T was set to 0.4 s. For each case, the optimized trajectory reduces the initial error. However, as illustrated by the decreasing white areas, the rate of convergence slows down for increasing values of c_r . This is also shown in Tab. 4.2, which summarizes the optimized costs J for c_r in the range from 0.60 to 0.75.

4.5.2 Dribbling Task

The schematic setup and the evolution chart of the dribbling task with a rigid end effector is illustrated in Fig. 4.5. The fundamental difference to the juggling task is the existence of an additional, autonomous impact that occurs between ball and ground. Due to this additional impact, a periodic solution for the dribbling task is defined by x_M^* , \dot{x}_M^* , and \dot{x}_B^* . For the juggling task, x_M^* and \dot{x}_M^* sufficed to define a periodic solution, compare Subsec.4.5.1.

Dynamic model. Again, a Poincaré cross section directly before the actuated impact is considered and the superscript – for the state is omitted. The position of ball and manipulator at the cross section are then related via

$$x_{B[n]} = x_{M[n]} - d_B/2 \quad (4.25)$$

and the state dimension is reduced to three. The time of the ground impact is denoted by t_{gi} and $\dot{x}_{B,gi[n]}^+ = \dot{x}_{B[n]}(t_{gi}^+)$ refers to the ball velocity after the n-th ground impact. With the state vector introduced in (4.15), the discrete-time dynamic model of the system

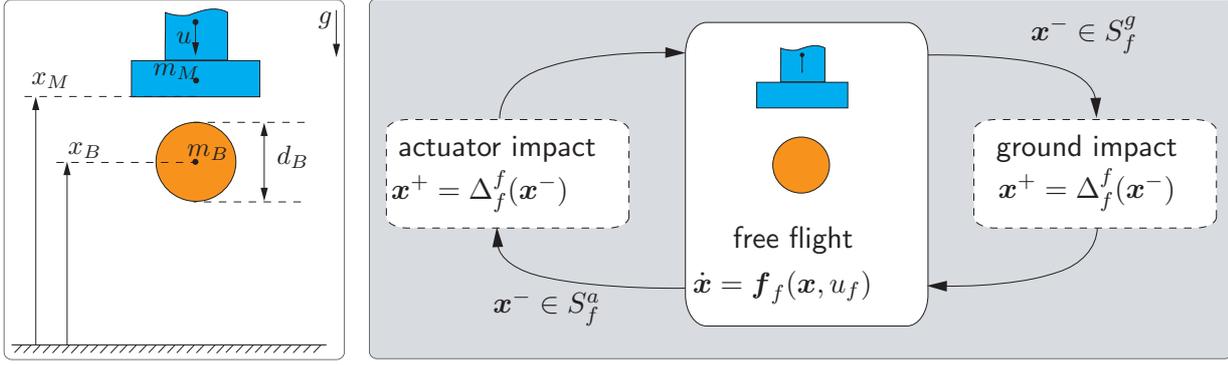


Fig. 4.5: Dribbling task: schematic (left) and evolution chart (right) of the dribbling system with a rigid actuator. Solid boxes depict discrete states, dashed boxes depict instantaneous transitions.

is given by integration of the continuous-time dynamics

$$\begin{aligned}
 \theta_{[n+1]} &= \theta_{[n]} + T\omega \\
 \dot{x}_{M[n+1]} &= \omega f'(\theta_{[n+1]}) \\
 \dot{x}_{B[n+1]} &= -gT_{rise} + \dot{x}_{B,gi[n]}^+ = \\
 &= \sqrt{\left(\dot{x}_{B,gi[n]}^+\right)^2 - 2gf(\theta_{[n+1]})} = \sqrt{c_r^2 \left(-gT_{drop} + \dot{x}_{B[n]}^+\right)^2 - 2gf(\theta_{[n+1]})} \\
 &= \sqrt{c_r^2 \left((1 + c_r)\dot{x}_{M[n]} - c_r\dot{x}_{B[n]}\right)^2 + 2gc_r^2 \left(f(\theta_{[n]}) - d_B\right) - 2gf(\theta_{[n+1]})}
 \end{aligned} \tag{4.26}$$

with the following abbreviations:

$$\begin{aligned}
 \dot{x}_{B[n]}^+ &= (1 + c_r)\dot{x}_{M[n]} - c_r\dot{x}_{B[n]}, & \dot{x}_{B,gi[n]}^+ &= -c_r\dot{x}_{B,gi[n]}^- = -c_r(-gT_{drop} + \dot{x}_{B[n]}^+), \\
 T_{drop} &= \frac{1}{g} \left(\dot{x}_{B[n]}^+ + \sqrt{\left(\dot{x}_{B[n]}^+\right)^2 + 2g(x_{M[n]} - d_B)} \right), \\
 T_{rise} &= \frac{1}{g} \left(\dot{x}_{B,gi[n]}^+ - \sqrt{\left(\dot{x}_{B,gi[n]}^+\right)^2 - 2gx_{M[n+1]}} \right).
 \end{aligned} \tag{4.27}$$

Local stability analysis. For the local stability analysis, the system (4.26) is also linearized around the fixed point using the implicit function theorem. The linearization of the Poincaré map $\left. \frac{\partial \mathbf{P}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*}$ can be derived analytically and results in a long symbolic expression. For the dribbling task, the linearized matrix and its eigenvalues are a function of the chosen fixed point \mathbf{x}^* . This is in contrast to the juggling task, where the eigenvalues were not influenced by the fixed point \mathbf{x}^* , compare (4.18)-(4.19). Due to this increased complexity, it is not possible to derive an analytical expression for the range of locally stable periodic orbits for the dribbling task. However, it is possible to perform a numerical analysis which evaluates the stability properties for different hybrid limit cycles. For the dribbling task, a periodic solution is defined by x_M^* , \dot{x}_M^* , and \dot{x}_B^* .

In the following analysis, the ball velocity before the actuated impact \dot{x}_B^* is constrained

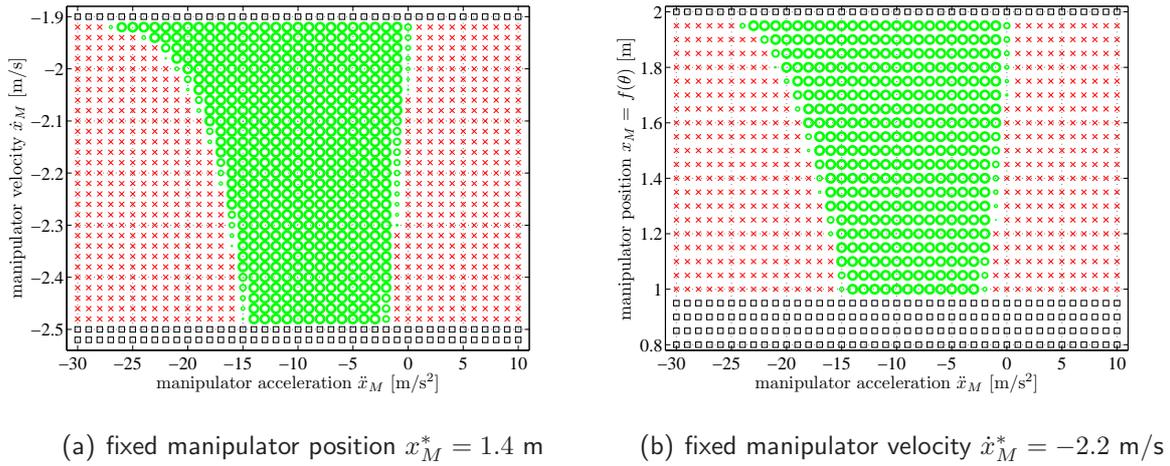


Fig. 4.6: Local stability of periodic solutions of the dribbling task for $c_r = 0.8$. Green circles denote locally stable combinations where the magnitude of the largest eigenvalue is smaller than one. A large marker corresponds to a small maximum eigenvalue. Red crosses denote unstable solutions and black squares illustrate infeasible solutions.

to the range $[0, 5]$ m/s. If there exist two feasible ball velocities within the range for a given combination of x_M^* and \dot{x}_B^* , the velocity with the smaller absolute value is chosen. The coefficient of restitution is assumed to be $c_r = 0.8$. With these assumptions, a local stability analysis for different periodic solutions can be performed by varying the manipulator position x_M^* and velocity \dot{x}_M^* at impact. Fig. 4.6 illustrates the relationships between these two quantities, the manipulator acceleration \ddot{x}_M and the magnitude of the largest eigenvalue for the linearized system. Fig. 4.6(a) shows combinations of manipulator accelerations and velocities for a fixed position $x_M^* = 1.4$ m. Fig. 4.6(b) shows combinations of manipulator accelerations and positions for a fixed manipulator velocity $\dot{x}_M^* = -2.2$ m/s. In both cases, local stability of a periodic orbit requires a negative manipulator acceleration. Furthermore, the lower and upper bounds for the manipulator acceleration depend on the hybrid limit cycle.

4.5.3 Comparison of Classic Juggling and Dribbling

The following paragraphs compare the classic juggling task with the dribbling task. A detailed evaluation of the differences is provided by Bätz et al. in [154].

Feasible periodic motions. As illustrated in Fig. 4.2, the hybrid limit cycle of the juggling task includes only one impact event, namely the actuation impact. For the dribbling task, the hybrid limit cycle includes two impact events: the actuation impact and the additional autonomous impact, compare Fig. 4.5. Consequently, for the juggling task, the ball trajectory of a periodic motion is unambiguously defined by x_M^* and \dot{x}_M^* or, alternatively, by its *energy level* E^* . For the dribbling task, in contrast, an infinite amount of feasible periodic orbits exists for a given energy level E^* : the hybrid limit cycle is defined by x_M^* , \dot{x}_M^* , and \dot{x}_B^* and hence, by varying the dribbling height x_M^* , the ratio between potential

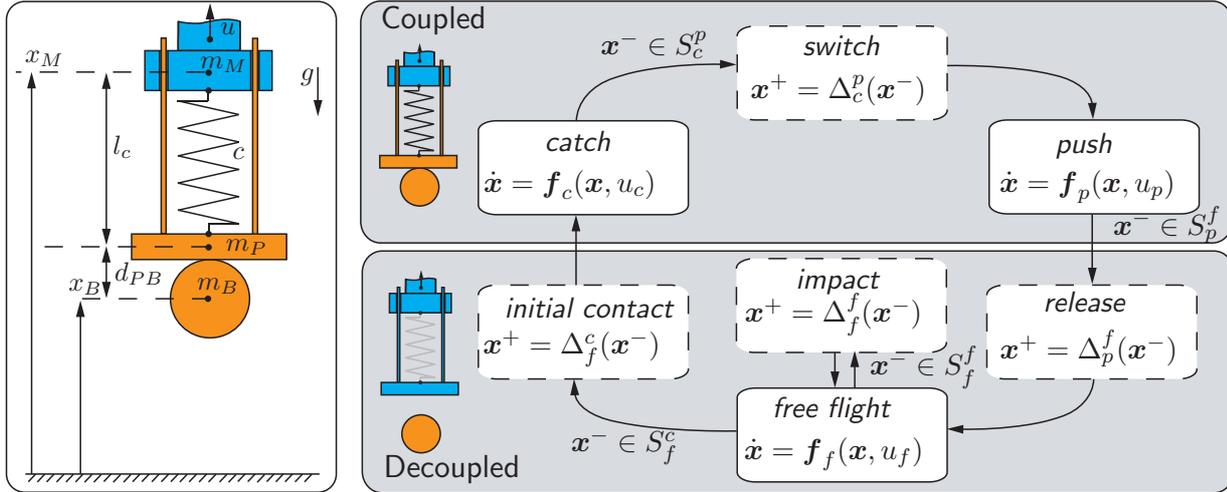


Fig. 4.7: Dribbling task: schematic (left) and evolution chart (right) of the hybrid ball dribbling system with a compliant end effector. Solid boxes depict discrete states, dashed boxes depict instantaneous transitions.

and kinetic energy at impact with the actuator is changed the resulting motion of a certain energy level is modified.

Local stability. For the juggling task, the linearized Poincaré map and its eigenvalues are independent of the desired fixed point \mathbf{x}^* , compare (4.18)-(4.19). Hence, local stability is only influenced by the manipulator acceleration \ddot{x}_M^* and the coefficient of restitution c_r . For the dribbling task, in contrast, the desired fixed point \mathbf{x}^* influences the linearization of the Poincaré map and hence local stability, compare Fig. 4.6.

For both tasks, the manipulator acceleration \ddot{x}_M^* at impact has no influence on the shape of the periodic motion, compare Subsec. 4.5.1 and 4.5.2. However, the manipulator acceleration influences the stability of the hybrid limit cycle: for both tasks, there exists a negative acceleration range in which periodic solutions are locally stable.

4.6 Application for Compliant End Effectors

The previous section discussed two periodic manipulation tasks for a rigid end effector design. The following section considers the dribbling task with a compliant end effector design. The contact between end effector and ball is still intermittent, however, the use of a spring element introduces a continuous-time control phase instead of an instantaneous impact. This modification resembles the human approach where a rather flexible *arm-hand manipulator* is used. The schematic in Fig. 4.7 shows the two DOF ball-spring-manipulator system actuated at the mass m_M . The switching surfaces and reset maps are depicted in Tab. 4.3.

Switching surface	Reset map
$S_c^p = \dot{x}_B$	$\mathbf{x}^+ = \Delta_c^p(\mathbf{x}^-) = \mathbf{x}^-$
$S_p^f = x_M - x_B - d_{PB} - l_{c,0}$	$\mathbf{x}^+ = \Delta_p^f(\mathbf{x}^-) = \mathbf{x}^-$
$S_f^f = x_B - d_B/2$	$\mathbf{x}^+ = \Delta_f^f(\mathbf{x}^-) = [1 \ 1 \ 1 - c_r]^T \mathbf{x}^-$
$S_f^c = x_M - x_B - d_{PB} - l_{c,0}$	$\mathbf{x}^+ = \Delta_f^c(\mathbf{x}^-) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ 0 & \frac{m_P}{\tilde{m}_B} \quad \frac{m_B}{\tilde{m}_B} \end{bmatrix} \mathbf{x}^-$

Tab. 4.3: Dribbling task with compliant end effector: switching surfaces and reset maps.

Coupled dynamics. When the ball is in contact with the plate, the system dynamics are given by

$$\begin{aligned} m_M \ddot{x}_M &= u - c(x_M - x_B - d_{PB} - l_{c,0}) - m_M g \\ \tilde{m}_B \ddot{x}_B &= c(x_M - x_B - d_{PB} - l_{c,0}) - \tilde{m}_B g, \end{aligned} \quad (4.28)$$

where $\tilde{m}_B = m_B + m_P$ is the merged mass of ball and plate and g the gravitational acceleration. The spring with stiffness c is restricted to its working range

$$l_c = x_M - x_B - d_{PB}, \quad l_c \in [l_{c,min}, l_{c,0}]. \quad (4.29)$$

Decoupled dynamics. Here, it is assumed that manipulator and plate are rigidly connected when the ball is not in contact with the plate. In practice, this can be realized by an electro-mechanical lock. This assumption yields to the following free-flight dynamics

$$\begin{aligned} \tilde{m}_M \ddot{x}_M &= u - \tilde{m}_M g \\ \ddot{x}_B &= -g, \end{aligned} \quad (4.30)$$

where $\tilde{m}_M = m_M + m_P$ is the merged mass of manipulator and plate.

Hybrid dynamics. The cyclic ball dribbling task is of hybrid nature consisting of continuous phases c (catch), p (push) and f (free flight) separated by instantaneous transitions. The state of the system is given by

$$\mathbf{x} = [x_M \ x_B \ \dot{x}_M \ \dot{x}_B]^T \quad (4.31)$$

and the dynamic model for phase $i \in \{c, p, f\}$ is $\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, u_i)$. Here, phases p and c are described by (4.28), and phase f is described by (4.30). Switching surfaces S_i^j determine the occurrence of a transition from phase i to j , compare Sec. 2.2. The reset map for such a transition can be written as

$$\mathbf{x}^+ = \Delta_i^j(\mathbf{x}^-) \quad \text{for } \mathbf{x}^- \in S_i^j \quad (4.32)$$

where \mathbf{x}^- is the state directly before the transition, \mathbf{x}^+ is the state directly after the transition, and $(i, j) \in \{(f, c), (c, p), (p, f), (f, f)\}$.

The hybrid model of the complete system is given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}_i(\mathbf{x}, u_i) & \mathbf{x}^- &\notin S_i^j \\ \mathbf{x}^+ &= \Delta_i^j(\mathbf{x}^-) & \mathbf{x}^- &\in S_i^j \end{aligned} \quad (4.33)$$

with $(i, j) \in \{(f, c), (c, p), (p, f), (f, f)\}$. A summary of discrete states and transitions for a ball dribbling cycle is given in Tab. 4.3.

The ball dynamics between release and the subsequent initial contact are not directly controlled. Hence, an algebraic relation for the state of the ball can be derived by integrating (4.30). Furthermore, it is assumed that the manipulator dynamics are fast enough to reach the desired state at *initial contact*, starting from *release* during the time that the ball falls to the ground and bounces back. For the system analysis, it is reasonable to consider only the controlled phase of the ball dribbling cycle involving the dynamics of the underactuated ball-spring-manipulator system (4.28). The time from *release* to ground *impact* is given by

$$t_f^f - t_p^f = \frac{1}{g} \left(\dot{x}_{B,p} + \sqrt{\dot{x}_{B,p}^2 + 2g(x_{B,p} - d_B/2)} \right), \quad (4.34)$$

where $\dot{x}_{B,p} = \dot{x}_B(t_p^f)$ and $x_{B,p} = x_B(t_p^f)$. The ball velocity after the ground impact can be described as function of the ball state at *release* or at *initial contact*:

$$\begin{aligned} \dot{x}_{B,f}^+ &= \dot{x}_B(t_f^{f+}) = c_r \sqrt{\dot{x}_{B,p}^2 + 2g(x_{B,p} - d_B/2)} \\ &= \sqrt{(\dot{x}_{B,c}^-)^2 + 2g(x_{B,c}^- - d_B/2)}, \end{aligned} \quad (4.35)$$

where $\dot{x}_{B,c}^- = \dot{x}_B(t_f^{c-})$ and $x_{B,c}^- = x_B(t_f^{c-})$. Using (4.34) and (4.35), the evolution chart in Fig. 4.7 can be modified by introducing a new mapping from *release* to *initial contact*.

Motion planning. During a dribbling cycle, there are two continuous phases that provide control over the ball, namely *catch* and *push*. The desired continuous-time motion of the ball-spring-manipulator system (4.28) can be described by the time evolution of its generalized coordinates

$$\{x_M = x_M^*(t), x_B = x_B^*(t)\}, \quad t \in [t_b, t_e], \quad t_b < t_e. \quad (4.36)$$

The facts that the dynamical system (4.28) is underactuated and the overall system (4.33) is of hybrid nature pose a challenge for the planning of desired motions.

Virtual holonomic constraints. By introducing a set of geometric relations among the general coordinates, the motion of (4.36) can be rewritten as

$$\{x_M = \phi_1(\theta), x_B = \phi_2(\theta)\}, \quad \theta = \theta^*(t), \quad t \in [t_b, t_e] \quad (4.37)$$

with a scalar variable $\theta \in [\theta_b, \theta_e]$ that is used as trajectory generator for parameterizing the time evolution. Geometric functions among the generalized coordinates as introduced

by (4.37) are known as *virtual holonomic constraints*, see Sec. 2.3. A convenient choice for the system is the following:

$$\begin{bmatrix} x_M \\ x_B \end{bmatrix} := \Phi(\theta) = \begin{bmatrix} \phi(\theta) \\ \theta \end{bmatrix}. \quad (4.38)$$

Suppose that there exists a control law $u^*(t)$ for the input u that makes the virtual holonomic constraint (4.38) invariant with respect to time. Then, the overall closed-loop system can be generally represented by reduced order dynamics of the form [123]

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0. \quad (4.39)$$

The solutions of this virtually constrained system define achievable motions with precise synchronization determined by (4.38). It means the whole motion is parameterized by the evolution of the chosen configuration variable θ . The smooth coefficient functions of (4.39) can be derived by substituting (4.38) into the system dynamics (4.28):

$$\alpha(\theta) = \tilde{m}_B, \quad \beta(\theta) = 0, \quad \gamma(\theta) = -c(\phi(\theta) - \theta - d_{PB} - l_{c,0}) + \tilde{m}_B g. \quad (4.40)$$

The reduced order dynamics of the form (4.39) is always integrable [123]. Specifically, the integral function

$$I(\theta_b, \dot{\theta}_b, \theta, \dot{\theta}) = \dot{\theta}^2 - \dot{\theta}_b^2 + \int_{\theta_b}^{\theta} \frac{2\gamma(s)}{\alpha(s)} ds \quad (4.41)$$

preserves its zero value along the solution of (4.39), initiated at $[\theta_b(t_b), \dot{\theta}_b(t_b)] = [\theta_b, \dot{\theta}_b]$.

With the above arguments, the motion planning problem is converted from a search of feasible orbits in the state space into a search for a parameterizing function $\phi(\theta)$ such that a desired solution of the reduced dynamics is found. Here, a Bézier polynomial

$$\phi(\theta) = \sum_{k=0}^M a_k \binom{M}{k} s^k (1-s)^{M-k}, \quad s = \frac{\theta - \theta_b}{\theta_e - \theta_b} \quad (4.42)$$

is used as geometric relation between the generalized coordinates [143]. Hence, one needs to find the coefficients $\mathbf{a} = [a_0 \dots a_M]$ that lead to the desired time evolution $\theta^*(t)$ between the specified initial and final conditions $[\theta_b, \dot{\theta}_b]$ and $[\theta_e, \dot{\theta}_e]$.

Trajectory optimization. For the trajectory optimization, the following restrictions are considered: at the beginning of the catching phase and at the end of the pushing phase, the spring is constrained to be at equilibrium length. Furthermore, the condition of continuity requires $\mathbf{x}^+ = \Delta_c^p(\mathbf{x}^-) = \mathbf{x}^-$. With these requirements, the Bézier coefficients $a_0^{(c)}$, $a_0^{(p)}$ and $a_M^{(p)}$ are fixed. The remaining coefficients $\mathbf{a}^{(c)}$ and $\mathbf{a}^{(p)}$ of the Bézier polynomials for the *catching* and the *pushing* phase are determined by an optimization. The nonlinear

optimization problem is defined as

$$\begin{aligned} \min_{\gamma \in \Gamma} \quad & J_i(\gamma) \\ \text{with } \Gamma = \quad & \{\gamma \in \mathbb{R}^{2M-1}, \mathbf{c}_i(\gamma) \leq \mathbf{0}, c_e(\gamma) = 0\} \\ \gamma = \quad & \left[a_1^{(c)} \dots a_M^{(c)}, a_1^{(p)} \dots a_{M-1}^{(p)} \right]. \end{aligned} \quad (4.43)$$

The inequality constraints $\mathbf{c}_i(\gamma)$ originate from the limitations of the physical system, such as maximum actuator acceleration and limits of the spring deflection. The equality constraint $c_e(\gamma)$ ensures the validity of the reduced order dynamics:

$$\mathbf{c}_i(\gamma) = \begin{bmatrix} \|u(\gamma)\| - a_{max} \\ (x_M(\gamma) - x_B - d_{PB}) - l_{c,0} \\ l_{c,min} - (x_M(\gamma) - x_B - d_{PB}) \end{bmatrix}, \quad c_e(\gamma) = I(\theta_b, \dot{\theta}_b, \theta, \dot{\theta}, \gamma). \quad (4.44)$$

Two different cost functions are considered for the trajectory optimization. The function J_1 evaluates the maximum manipulator velocity and J_2 estimates the energy consumption:

$$\begin{aligned} J_1(\gamma) &= \lambda \max \|\dot{x}_M(t)\|^2 + e(t_e) \\ J_2(\gamma) &= \frac{\lambda}{T} \int_{t_b}^{t_e} u(t)^2 dt + e(t_e), \end{aligned} \quad (4.45)$$

with $\lambda = 10^2$, $T = t_e - t_b$ and the terminal cost

$$e(t_e) = \lambda_v e_v(t_e)^2 + \lambda_p e_p(t_e)^2 \quad (4.46)$$

which punishes deviations of the final velocity and position ($\lambda_v = \lambda_p = 10^6$) at the end of each phase. Based on these cost functions, trajectory optimization has been performed for different degrees of the Bézier polynomial (4.42). For the optimization results, a particular ball dribbling cycle is considered: the desired dribbling height is set to $h_d = x_{B,c}^- = x_{B,p}^+ = 0.9$ m, the *initial contact* to $[x_{B,c}^+, \dot{x}_{B,c}^+] = [0.8 \text{ m}, 1.72 \text{ m/s}]$ and the *release* to $x_{B,p}^- = x_{B,f}^+ = 0.5$ m is considered. With a known coefficient of restitution c_r , the ball velocity $\dot{x}_{B,p}^-$ at release is determined with (4.35). Table 4.4 summarizes the results of the numerical optimization. For both cost functions, the higher order Bézier polynomials naturally lead to reduced costs. However, there is no significant difference in the costs between $M = 5$ and $M = 6$. Hence, a Bézier polynomial of order $M = 5$ is a good trade-off between cost optimality and numerical complexity, which is relevant for the implementation on the real system. For comparison, Fig. 4.8 shows optimized trajectories and the resulting spring deflection for $M = 3$ and $M = 5$ obtained when using J_1 .

Control design: transverse linearization. Given the desired orbit obtained through the optimization process, the next step is to design a controller that stabilizes the ball-spring-manipulator system in the vicinity of the desired limit cycle. Based on the system description in the form of (4.37), new coordinates and velocities are introduced in the

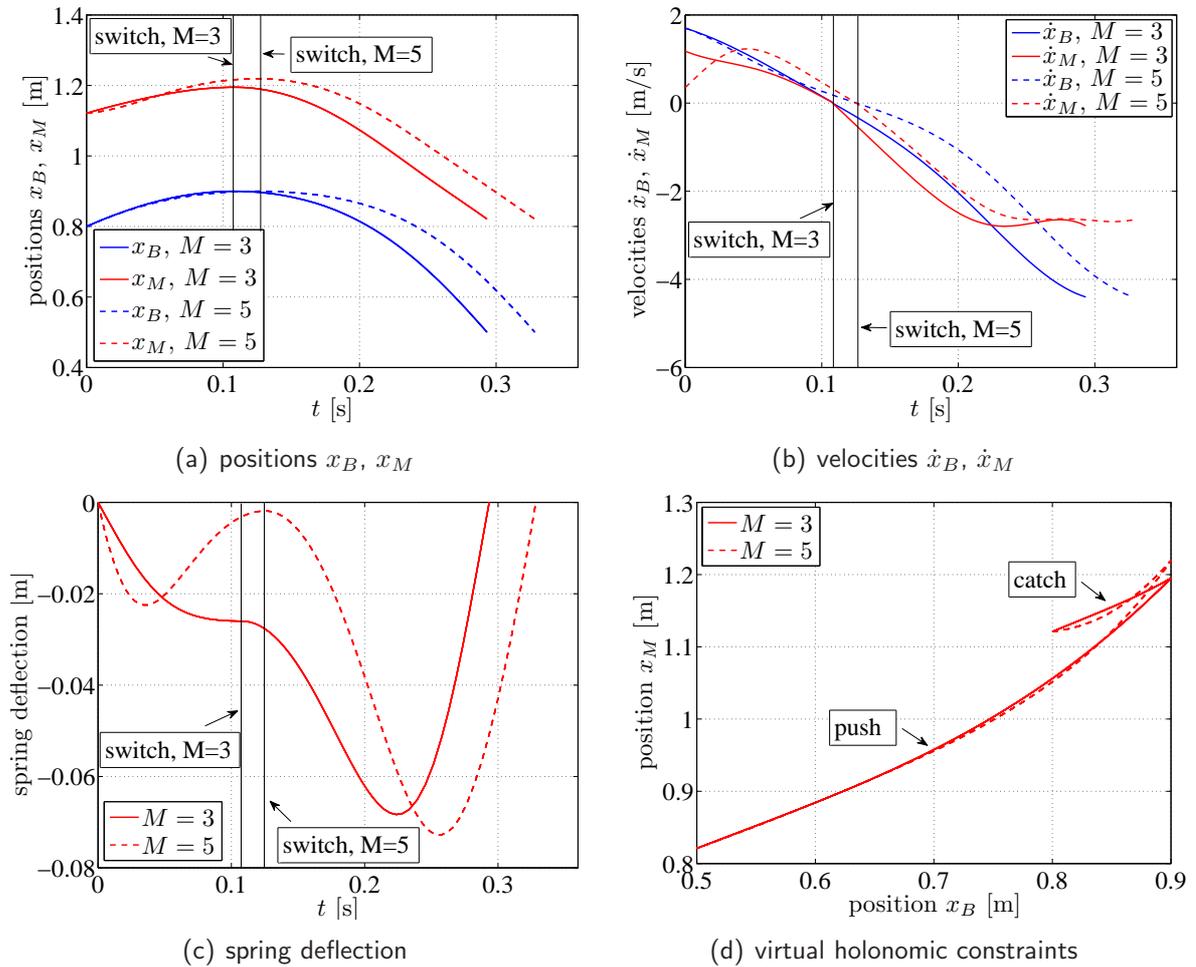


Fig. 4.8: Optimized trajectories for different Bézier degrees using cost function J_1 . The solid lines indicate the switch from *catch* to *push* phase.

Cost function	Bezier degree			
	3	4	5	6
J_1	779	720	708	706
J_2	80249	77663	75234	74134

Tab. 4.4: Costs of the optimized trajectories for different Bézier degrees.

vicinity of the target orbit:

$$\begin{aligned} x_M &= \phi(\theta) + y, & x_B &= \theta \\ \dot{x}_M &= \phi'(\theta)\dot{\theta} + \dot{y}, & \dot{x}_B &= \dot{\theta}. \end{aligned} \quad (4.47)$$

The dynamics of the synchronization error y to the specified virtual holonomic constraint can be computed from the system dynamics (4.28) by substituting (4.47):

$$\ddot{y} = r(\theta, \dot{\theta}, y, \dot{y}) + (1/m_M)u = v. \quad (4.48)$$

An auxiliary control signal v is introduced by a control transformation via partial feedback linearization [131]. The dynamics of θ is given by substituting (4.47) into the second order differential equation (4.39), which yields

$$\tilde{m}_B \ddot{\theta} - c(\phi(\theta) - \theta - d_{PB} - l_{c,0}) + \tilde{m}_B g = -cy, \quad (4.49)$$

where the right hand side equals to zero on the desired orbit. The target motion (4.36) in the new generalized coordinates is given by

$$\{y^*(t) \equiv 0, \theta = \theta^*(t)\}, \quad t \in [t_b, t_e]. \quad (4.50)$$

The dynamic system has a natural choice of transverse coordinates

$$\mathbf{x}_\perp = [I(\theta, \dot{\theta}, \theta_b^*, \dot{\theta}_b^*) \quad y \quad \dot{y}]^T, \quad (4.51)$$

that describe the system's behavior away from a specified orbit [123]. A transverse linearization along a continuous-time target motion (4.50) can be analytically computed to be used for system analysis and control design:

$$\dot{\mathbf{z}} = \mathbf{A}(t)\mathbf{z} + \mathbf{b}v = \begin{bmatrix} 0 & \frac{2\dot{\theta}^*(t)}{\tilde{m}_B}c & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I \\ Y_1 \\ Y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v, \quad (4.52)$$

where $\mathbf{z} = [I \quad Y_1 \quad Y_2]^T$ denotes the linearization for \mathbf{x}_\perp . A cyclic solution $\mathbf{z} = \mathbf{z}(t) = \mathbf{z}(t+T)$ with time period $T = t_p^f - t_f^c$ and a switch at $t_s = t_c^p$ is defined by

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{A}(s)\mathbf{z} + \mathbf{b}v, \quad s = t \bmod T \\ \mathbf{A}(s) &= \begin{cases} \mathbf{A}_c(s), & s \in (0, t_s] \\ \mathbf{A}_p(s), & s \in (t_s, T) \end{cases} \\ \mathbf{z}(t_k^+) &= \mathbf{f}_{ri}(\mathbf{z}(t_k^-)), \quad t_k = kT, \quad k \in \mathbb{N}. \end{aligned} \quad (4.53)$$

The operator \mathbf{f}_{ri} is the mapping from *release* to *initial contact* based on (4.35).

Closed-loop system. Consider the control law

$$v = \mathbf{k}(s)\mathbf{z}, \quad \mathbf{k}(s) = \begin{cases} \mathbf{k}_c(s), & s \in (0, t_s] \\ \mathbf{k}_p(s), & s \in (t_s, T) \end{cases} \quad (4.54)$$

for the hybrid linear system (4.53). It can be shown that exponential stability of the origin for the linear impulsive system is equivalent to exponential orbital stability of the periodic motion for the nonlinear one [124]. For reasons of implementation, only a constant feedback of the synchronization error $x_M - \phi(x_B)$ is used since these measurements are available:

$$\mathbf{k}_c = [0 \quad k_{c2} \quad 0], \quad \mathbf{k}_p = [0 \quad k_{p2} \quad 0]. \quad (4.55)$$

In addition to the control during the two contact phases *catch* and *push*, a control for the manipulator during the non-contact phase has to be applied. This control law must ensure that the manipulator is at the desired state when the controlled phase of the ball dribbling cycle starts, compare Subsec. 6.4.2.

4.7 Summary

The chapter discussed periodic manipulation tasks with intermittent contact. In Section 4.1, an overview on related work was given and the challenges of the trajectory planning were addressed in Section 4.2. Next, methods for the stability analysis of periodic solutions of (hybrid) dynamical systems were outlined in Sections 4.3 and 4.4. Then, two applications for rigid end effectors were presented in Section 4.5. First, the classic juggling task was studied: the section reviewed the control strategies proposed in the literature and presented a new approach for optimal trajectory planning based on a non-local stability analysis. Ball dribbling was introduced as novel case study for dynamic dexterity. The system dynamics and optimal trajectory planning were discussed. A comparison with the classic juggling task showed that the dribbling task is more challenging due to the occurrence of an additional, autonomous impact. In addition to dribbling with a rigid end effector, a novel approach for the stabilization of human-like periodic ball dribbling motion was presented in Section 4.6. Adding an elastic element to the manipulator promises two benefits: first, the ball can be controlled in a continuous-time phase instead of an intermittent contact. Second, impacts between manipulator and ball are avoided which reduces the mechanical load. The virtual holonomic constraints approach was used to plan optimal catching and pushing trajectories considering two different cost functions. An orbitally stabilizing feedback controller was designed for the underactuated ball-spring-manipulator system based on a transverse linearization along the desired motion.

5 Perception and Interaction Control for Dynamic Manipulation

Summary This chapter addresses the particular challenges that arise for the closed-loop control of dynamic manipulation tasks and presents an extensive control structure to overcome these challenges. Four different observer designs are developed to reconstruct the forces/torques that are exchanged between the manipulator and its environment during dynamic motions. The designs are evaluated in a simulation scenario as well as an experiment. The force/torque observer is then integrated in a direct force control scheme to improve the interaction control for dynamic manipulation tasks. Finally, an approach for high-speed image processing is presented which allows to track and to predict the state of objects during non-contact phases with high sampling rates.

The two previous chapters discussed trajectory planning and control strategies for dynamic manipulation tasks. It was shown that the successful execution of these tasks generally requires some amount of sensory feedback. Clearly, state feedback is also needed when performing conventional manipulation tasks. However, the dynamic nature of the tasks studied in this work poses some additional challenges which have to be overcome. The chapter addresses these challenges and the obtained results are prerequisites for the experiments presented in Chapter 6.

In principle, feedback can be provided either at discrete times or continuously throughout the task execution. During non-contact phases, the most valuable source of feedback is visual information. When the manipulator is in contact with the environment, force/torque measurements are another essential source of information. Like visual feedback, they allow to determine the pose of the manipulated object. Additionally, the measured forces and torques also provide information about the acceleration of the object. This information, however, is not directly accessible since the force/torque measurements are disturbed by inertial and gravitational forces and measurement noise. For dynamic motions, this is particularly a problem, since the tool inertia has here a non-negligible effect on the measurements of the wrist F/T-sensor. Due to this fact, the chapter discusses observer designs which allow the reconstruction of the forces and torques exchanged between the manipulator and the environment.

Accurate knowledge of these contact forces improves the performance of the applied interaction control scheme (direct or indirect force control). In addition to this sensory information, a precise underlying motion control of the robot is a prerequisite for successful

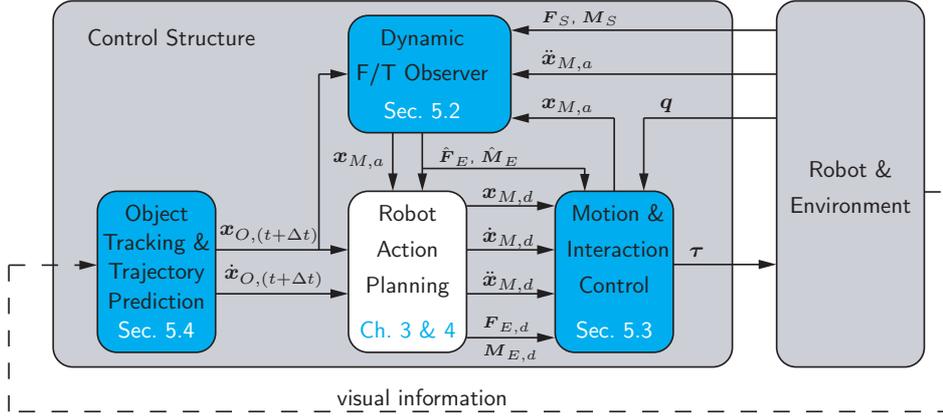


Fig. 5.1: Overall control structure highlighting the elements that are addressed in this chapter: dynamic F/T observer, object tracking & trajectory prediction, and motion & interaction control.

task execution. Typically, this is realized with an inverse dynamics control scheme.

The outline of the chapter is as follows: Section 5.1 provides an overview on related work. In addition to works in the field of interaction control, the section details research efforts in high-speed vision processing and in dynamic force/torque sensing. Section 5.2 discusses observer designs for the reconstruction of the forces and torques exchanged with the environment. Four different designs are evaluated in a simulation scenario and in experiments. Then, Section 5.3 focuses on robot motion and interaction control schemes that are suitable for dynamic manipulations tasks. Besides a classical inverse dynamics control, a parallel force/position control scheme is reviewed. Finally, Section 5.4 details the vision-based object tracking and trajectory prediction. For the image processing, the challenge is finding the optimal trade-off between two competing goals: high tracking accuracy and high frame rates. The overall control structure is shown in Fig. 5.1 and the elements addressed in this chapter are highlighted.

5.1 Related Work

The knowledge of contact forces and torques is particularly crucial for the interaction with the environment. For dynamic manipulation tasks, the extraction of these contact forces is a challenging task. This problem has been addressed by various researchers: Dynamic force sensing for high-speed robot manipulation was first investigated by Uchiyama et al. [139]. The developed observer could estimate both environment forces and torques and was based on the Extended Kalman Filter. The design, however, was restricted to pose and F/T measurements only. In the presented results, the complexity of the problem was reduced by considering only a planar scenario with two translational and one rotational DOF.

Lin developed an observer based on position and force measurements in [78]. In this work, only environment forces and translational motions were considered which results in linear process and measurement equations. Hence, a classical Kalman Filter could be applied to estimate the contact forces.

Garcia et al. investigated a sensor fusion approach for dynamic force/torque estimation [42, 43]. In addition to pose and F/T measurements, an inertial sensor was used to measure the tool acceleration. In the EKF-based observer design, the environment forces and torques were not considered. The estimation error was utilized to obtain a contact F/T estimator with low-pass properties. Considering the nonlinear process model for the torques, it seems problematic to use the estimation error of such an observer to determine the environment torques. In addition, the representation of the tool orientation with Euler angles appears critical because of the well-known issues with representation singularities.

Kröger et al. also discussed a fusion of F/T and acceleration information for contact force/torque estimation [74]. In their work, characteristic properties of F/T and acceleration sensors were reported [75]. Furthermore, translational and rotational acceleration estimates based on pose measurements were derived. However, the calculation of contact forces and torques was performed without considering the stochastic properties of the signals, hence limiting the effectiveness of the approach.

For dynamic manipulation tasks, the precise timing and execution of motions are preliminary requirements. Inverse dynamics control schemes are a well-established method for improved motion control [126, 125, 94]. Friction compensation is an essential part of the inverse dynamics control. Due to the various friction phenomena that need to be considered, this is a challenging task (compare Sec. 2.5). Surveys on friction compensation for robotic manipulators were provided by Olsson and Bona [96, 97, 14]. Detailed experimental results on friction compensation for the dual-arm robot used in this work have been presented by Ueberle [140]. Comprehensive discussions on motion control for robotic manipulators were presented by Siciliano and Khatib [126, 125].

In addition to precise motion execution, dynamic manipulation tasks also require interaction with the environment. Hence, an interaction control is needed in addition to pure motion control. Numerous strategies for such an interaction control have been proposed in literature: in general, these interaction control schemes can be classified into direct and indirect force control. The former approach, which includes impedance and admittance control, was first proposed by Hogan [53]. A drawback of indirect force control schemes is the fact that the contact forces/torques at the end effector can not be directly controlled. This shortcoming is addressed by direct force control, which allows to control the forces/torques that are exchanged between end effector and environment. Direct force control was first proposed by Raibert and Craig and later extended by Yoshikawa and Khatib to include an inverse dynamics motion control [101, 149, 70]. These hybrid force/position control schemes use a task-specific selection matrix that determines for each task coordinate whether it is force or position controlled. This is a drawback as the structure of the control has to be changed during task execution. The problem was addressed by Chiaverini and Sciavicco, who presented a *parallel* approach to force/position control of robotic manipulators [28]. Here, the position and force controller act in parallel, and conflicting situations are resolved by assigning a higher priority to the force control. The approach was extended by Natale for interaction control of six DOF tasks [94]. A comprehensive survey on interaction control schemes including experimental results was presented by Chiaverini et al. and Natale [29, 94]. For the intended applications in this thesis, the parallel approach is best suited as it offers both direct force control and relatively high

structural flexibility.

Rizzi and Koditschek were the first who emphasized the importance of visual attention control for dynamically dexterous robots [106]. In order to achieve higher frame rates, they proposed to reduce the search window on the images obtained from the camera system. High-speed vision improves control performance in vision-based control due to higher frame rates and lower latencies. Recent progress in sensors, bus systems and semiconductor technology has led to a few works on vision with more than 30 Hz frame rate during the last decade: Ishikawa et al. have developed high-speed vision systems with various resolutions and fast low-level image processing functions integrated on customized processing hardware allowing framerates of up to 1 kHz. These vision systems were applied to various dynamic vision-based manipulation tasks, e.g. ball dribbling with multi-fingered hands, ball catching, ball batting, and regrasping [122, 59, 119, 39]. In the field of visual servo control, a number of works with higher framerates (> 30 Hz) exist, see e.g. [93, 41, 146].

5.2 Dynamic Contact Force/Torque Observer

For robots that are destined to leave the classical industrial settings, one major challenge is the physical interaction with unknown and/or changing environments. Such an interaction requires knowledge of the exchanged contact forces and torques. To this end, robotic systems nowadays are typically equipped with force/torque sensors at the wrist. By using force control schemes that rely on measurements from these sensors, conventional manipulation tasks are successfully executed. In particular for dynamic manipulation tasks, however, the problem arises that the inertial forces/torques of the end effector have a non-negligible effect on the measurements of the wrist sensor. This does not only degrade the performance of the interaction control but also constitutes a safety risk.

The section discusses four different contact force/torque observer designs to overcome this problem: two approaches are based on the Extended Kalman Filter (EKF) and two approaches are based on the Unscented Kalman Filter (UKF). For each case, two different measurement vectors are considered: the first one only uses pose and force/torque measurements, whereas the second one also uses acceleration measurements to determine the contact forces and torques. The four observer designs are evaluated in simulation and experiment for six degrees-of-freedom tasks, paying particular attention to the influence of changing operating conditions.

Fig. 5.2 illustrates the typical hardware setup with a F/T-sensor located at the wrist. For dynamic motions, the inertial forces/torques of the end effector have a non-negligible effect on the force sensor measurements. Hence, the measurements do not correspond to the actual environmental interaction forces/torques. In general, the measured forces \mathbf{F}_S and torques \mathbf{M}_S are constituted by the following components

$$\begin{bmatrix} \mathbf{F}_S(t) \\ \mathbf{M}_S(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_E(t) \\ \mathbf{M}_E(t) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_I(t) \\ \mathbf{M}_I(t) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_G(t) \\ \mathbf{M}_G(t) \end{bmatrix} \quad (5.1)$$

where \mathbf{F}_E and \mathbf{M}_E are the environmental contact forces and torques, \mathbf{F}_I and \mathbf{M}_I are the inertial forces and torques, and \mathbf{F}_G and \mathbf{M}_G are the forces and torques due to gravity.

Let C denote the center of mass of the end effector and Σ_C the body-fixed coordinate frame with origin in C and which axes coincide with the principal axes of inertia of the end effector. Using the center of mass C as the reference point, the Newton-Euler equations for the end effector with respect to frame Σ_C (illustrated by the superscript C) are given by

$$\begin{aligned} \mathbf{F}_S^C - \mathbf{F}_E^C + m\mathbf{g}^C &= m\mathbf{a}^C \\ \mathbf{S}(\mathbf{r}_{CS}^C)\mathbf{F}_S^C + \mathbf{M}_S^C - \mathbf{S}(\mathbf{r}_{CE}^C)\mathbf{F}_E^C - \mathbf{M}_E^C &= \mathbf{J}_C^C\boldsymbol{\alpha}^C + \mathbf{S}(\boldsymbol{\omega}^C)\mathbf{J}_C^C\boldsymbol{\omega}^C \end{aligned} \quad (5.2)$$

Here, the tool mass is given by m , \mathbf{J}_C^C is the moment of inertia of the tool with respect to C , \mathbf{a}^C denotes the translational end effector acceleration, $\boldsymbol{\omega}^C/\boldsymbol{\alpha}^C$ the angular end effector velocity/acceleration, and \mathbf{g}^C the vector of gravitational acceleration. The matrix \mathbf{R}_j^i is the rotation matrix of frame Σ_j with respect to frame Σ_i .

It should be noted that most quantities in (5.2) are measured (calculated) in their own frame. Thus, an extended formulation of the Newton-Euler equations for the actual implementation is given as

$$\begin{aligned} \mathbf{R}_S^C\mathbf{F}_S^S - \mathbf{R}_E^C\mathbf{F}_E^E + m\mathbf{R}_W^C\mathbf{g}^W &= m\mathbf{R}_I^C\mathbf{a}^I \\ \mathbf{S}(\mathbf{r}_{CS}^C)\mathbf{R}_S^C\mathbf{F}_S^S + \mathbf{R}_S^C\mathbf{M}_S^S - \mathbf{S}(\mathbf{r}_{CE}^C)\mathbf{R}_E^C\mathbf{F}_E^E - \mathbf{R}_E^C\mathbf{M}_E^E &= \mathbf{J}_C^C\mathbf{R}_I^C\boldsymbol{\alpha}^I + \mathbf{S}(\mathbf{R}_I^C\boldsymbol{\omega}^I)\mathbf{J}_C^C\mathbf{R}_I^C\boldsymbol{\omega}^I. \end{aligned} \quad (5.3)$$

In matrix notation, one obtains

$$\begin{aligned} \begin{bmatrix} \mathbf{F}_{S \rightarrow C}^C \\ \mathbf{M}_{S \rightarrow C}^C \end{bmatrix} - \begin{bmatrix} \mathbf{F}_{E \rightarrow C}^C \\ \mathbf{M}_{E \rightarrow C}^C \end{bmatrix} &= \\ &= \begin{bmatrix} m\mathbf{R}_I^C & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_C^C\mathbf{R}_I^C \end{bmatrix} \begin{bmatrix} \mathbf{a}^I \\ \boldsymbol{\alpha}^I \end{bmatrix} + \begin{bmatrix} -m\mathbf{R}_W^C & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\mathbf{R}_I^C\boldsymbol{\omega}^I)\mathbf{J}_C^C\mathbf{R}_I^C \end{bmatrix} \begin{bmatrix} \mathbf{g}^W \\ \boldsymbol{\omega}^I \end{bmatrix}. \end{aligned} \quad (5.4)$$

where the expressions $\mathbf{F}_{j \rightarrow C}^C$, $\mathbf{M}_{j \rightarrow C}^C$ with $j \in \{S, E\}$ correspond to

$$\begin{bmatrix} \mathbf{F}_{j \rightarrow C}^C \\ \mathbf{M}_{j \rightarrow C}^C \end{bmatrix} = \begin{bmatrix} \mathbf{R}_j^C & \mathbf{0} \\ \mathbf{S}(\mathbf{r}_{Cj}^C)\mathbf{R}_j^C & \mathbf{R}_j^C \end{bmatrix} \begin{bmatrix} \mathbf{F}_j^j \\ \mathbf{M}_j^j \end{bmatrix} \quad (5.5)$$

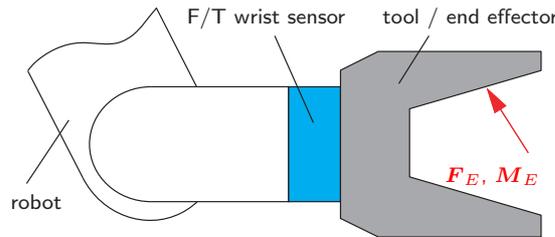


Fig. 5.2: Schematic of the hardware setup: the tool / end effector that interacts with the environment is attached to a wrist F/T sensor.

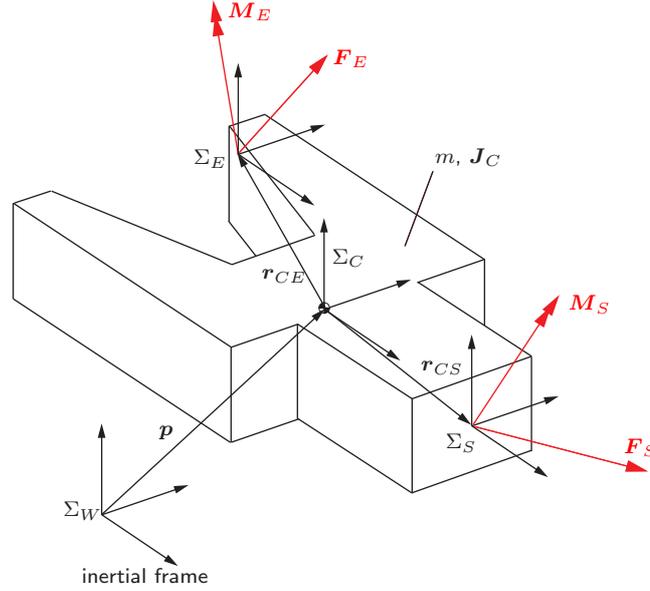


Fig. 5.3: Forces and moments applied at the end effector

5.2.1 Continuous-Time System Model

In the following, all quantities are expressed in frame C and in order to facilitate the notation, the superscript C is omitted. However, velocities and acceleration are, of course, absolute quantities calculated with respect to the inertial frame and only expressed in frame C. Let

$$\mathbf{x}(t) = \mathbf{x} = [\mathbf{p}^T \quad \mathbf{o}^T \quad \mathbf{v}^T \quad \boldsymbol{\omega}^T \quad \mathbf{F}_S^T \quad \mathbf{M}_S^T \quad \mathbf{F}_E^T \quad \mathbf{M}_E^T]^T \in \mathbb{R}^{25} \quad (5.6)$$

denote the state vector of the system. Here, \mathbf{p} and \mathbf{o} are the position and orientation (in quaternion) of the tool center of mass with respect to the inertial frame and \mathbf{v} is the (absolute) translational tool velocity. For the observer design, two different measurement vectors

$$\begin{aligned} \mathbf{y}_1(t) = \mathbf{y}_1 &= [\mathbf{p}^T \quad \mathbf{o}^T \quad \mathbf{F}_S^T \quad \mathbf{M}_S^T]^T \in \mathbb{R}^{13} \\ \mathbf{y}_2(t) = \mathbf{y}_2 &= [\mathbf{p}^T \quad \mathbf{o}^T \quad \mathbf{F}_S^T \quad \mathbf{M}_S^T \quad \mathbf{a}^T \quad \boldsymbol{\alpha}^T \quad \boldsymbol{\omega}^T]^T \in \mathbb{R}^{22} \end{aligned} \quad (5.7)$$

are considered. While \mathbf{y}_1 relies solely on position and force/torque measurements, \mathbf{y}_2 also includes acceleration and (rotational) velocity measurements. As proposed by [139], it is assumed that the process model for the sensor and environment forces and torques can be approximated by a white and Gaussian process. Based on (5.4)-(5.7), the nonlinear, continuous-time system model

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{w}) \\ \mathbf{y}_i &= \mathbf{h}_i(\mathbf{x}, \boldsymbol{\nu}) \end{aligned} \quad (5.8)$$

can be derived. The variables \mathbf{w} and $\boldsymbol{\nu}$ represent the process and measurement noise. They are assumed to be white, independent of each other, and with normal probability distributions

$$\begin{aligned} \mathbf{p}(\mathbf{w}) &\sim N(\mathbf{0}, \mathbf{Q}) \\ \mathbf{p}(\boldsymbol{\nu}) &\sim N(\mathbf{0}, \mathbf{R}), \end{aligned} \quad (5.9)$$

where the process and measurement noise covariances \mathbf{Q} , \mathbf{R} can vary with each measurement or time step.

5.2.2 Discrete-Time System Model

The nonlinear process and measurement model are given by the stochastic difference equation

$$\begin{aligned} \mathbf{x}_{[k+1]} &= \mathbf{f}(\mathbf{x}_{[k]}, \mathbf{w}_{[k]}), \\ \mathbf{y}_{i,[k+1]} &= \mathbf{h}_i(\mathbf{x}_{[k+1]}, \boldsymbol{\nu}_{[k+1]}), \end{aligned} \quad (5.10)$$

where the subscripts $[k+1], [k]$ denote the discrete time steps. The detailed difference equation for the process is given by

$$\begin{aligned} \mathbf{x}_{[k+1]} &= \mathbf{f}(\mathbf{x}_{[k]}, \mathbf{w}_{[k]}) = & (5.11) \\ &= \begin{bmatrix} \mathbf{p}_{[k]} + T\mathbf{v}_{[k]} + 0.5\frac{T^2}{m} (\mathbf{F}_{S[k]} - \mathbf{F}_{E[k]}) + \mathbf{w}_{p,[k]} \\ \mathbf{o}_{[k]} * \mathbf{o}_w * \mathbf{o}_\Delta \\ \mathbf{v}_{[k]} + \frac{T}{m} (\mathbf{F}_{S[k]} - \mathbf{F}_{E[k]} + \mathbf{w}_{v,[k]}) \\ \boldsymbol{\omega}_{[k]} + T(\mathbf{J}_C)^{-1} \left(\mathbf{S}_{r_{CS}} \mathbf{F}_{S[k]} - \mathbf{S}_{r_{CE}} \mathbf{F}_{E[k]} + \mathbf{M}_{S[k]} - \mathbf{M}_{E[k]} - \mathbf{S}_{\boldsymbol{\omega}_{[k]}} \mathbf{J}_C \boldsymbol{\omega}_{[k]} \right) + \mathbf{w}_{\boldsymbol{\omega},[k]} \\ \mathbf{F}_{S[k]} + \mathbf{w}_{F_S,[k]} \\ \mathbf{M}_{S[k]} + \mathbf{w}_{M_S,[k]} \\ \mathbf{F}_{E[k]} + \mathbf{w}_{F_E,[k]} \\ \mathbf{M}_{E[k]} + \mathbf{w}_{M_E,[k]} \end{bmatrix}. \end{aligned}$$

Here, $\mathbf{o}_w = \mathbf{w}_{o,[k]}$ is the quaternion reflecting process noise and

$$\mathbf{o}_\Delta = \{\eta_\Delta, \boldsymbol{\epsilon}_\Delta\} = \left\{ \cos\left(\frac{1}{2} \|\boldsymbol{\omega}_{[k]}\| T\right), \frac{\boldsymbol{\omega}_{[k]}}{\|\boldsymbol{\omega}_{[k]}\|} \sin\left(\frac{1}{2} \|\boldsymbol{\omega}_{[k]}\| T\right) \right\}$$

denotes the differential rotation during the time interval T [94, 73]. As the measurement equations for \mathbf{y}_1 are linear, they can be written in the following matrix notation

$$\begin{aligned} \mathbf{y}_{1,[k+1]} &= \mathbf{h}_1(\mathbf{x}_{[k+1]}, \boldsymbol{\nu}_{[k+1]}) = & (5.12) \\ &= \begin{bmatrix} \mathbf{I}_{7 \times 7} & \mathbf{0}_{7 \times 6} & \mathbf{0}_{7 \times 6} & \mathbf{0}_{7 \times 6} \\ \mathbf{0}_{6 \times 7} & \mathbf{0}_{6 \times 6} & \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix} \mathbf{x}_{[k+1]} + \boldsymbol{\nu}_{[k+1]} = \mathbf{H}_1 \mathbf{x}_{[k+1]} + \boldsymbol{\nu}_{1,[k+1]} \end{aligned}$$

Due to the incorporation of the angular acceleration, the measurement equations for \mathbf{y}_2

are nonlinear:

$$\begin{aligned} \mathbf{y}_{2,[k+1]} &= \mathbf{h}_2(\mathbf{x}_{[k+1]}, \boldsymbol{\nu}_{[k+1]}) = & (5.13) \\ &= \begin{bmatrix} \mathbf{H}_1 \mathbf{x}_{[k+1]} + \boldsymbol{\nu}_{1,[k+1]} \\ \frac{1}{m} (\mathbf{F}_{S[k+1]} - \mathbf{F}_{E[k+1]}) + \boldsymbol{\nu}_{a,[k+1]} \\ \mathbf{J}_C^{-1} \left(\mathbf{M}_{S[k+1]} - \mathbf{M}_{E[k+1]} - \mathbf{S}_{\omega_{[k+1]}} \mathbf{J}_C \boldsymbol{\omega}_{[k+1]} + \mathbf{S}_{r_{CS}} \mathbf{F}_{S[k+1]} - \mathbf{S}_{r_{CE}} \mathbf{F}_{E[k+1]} \right) + \boldsymbol{\nu}_{\alpha,[k+1]} \\ \boldsymbol{\omega}_{[k+1]} + \boldsymbol{\nu}_{\omega,[k+1]} \end{bmatrix}. \end{aligned}$$

5.2.3 Filter Design

As shown in (5.11), the system is characterized by a nonlinear process model. In addition, the measurement model for \mathbf{y}_2 is also nonlinear, compare (5.13). Hence, a nonlinear filter design is needed for the six DOF dynamic F/T observer. In the following, two different approaches for nonlinear systems are evaluated: Extended Kalman Filter and Unscented Kalman Filter.

Extended Kalman Filter (EKF). The EKF extends the Kalman Filter for nonlinear systems by linearizing about the current mean and covariance and can hence be applied to the given nonlinear estimation problem [142]. Using the following notations for the partial derivatives,

$$\begin{aligned} \mathbf{A}_{[k]} &= \frac{\partial \mathbf{f}(\hat{\mathbf{x}}_{[k-1]}, \mathbf{0})}{\partial \mathbf{x}} & \mathbf{W}_{[k]} &= \frac{\partial \mathbf{f}(\hat{\mathbf{x}}_{[k-1]}, \mathbf{0})}{\partial \mathbf{w}} \\ \mathbf{H}_{i,[k]} &= \frac{\partial \mathbf{h}_i(\hat{\mathbf{x}}_{[k-1]}, \mathbf{0})}{\partial \mathbf{x}} & \mathbf{V}_{i,[k]} &= \frac{\partial \mathbf{h}_i(\hat{\mathbf{x}}_{[k-1]}, \mathbf{0})}{\partial \boldsymbol{\nu}}, \end{aligned} \quad (5.14)$$

the EKF time update equations are given by

$$\begin{aligned} \hat{\mathbf{x}}_{[k]}^- &= \mathbf{f}(\hat{\mathbf{x}}_{[k-1]}^-, \mathbf{0}) \\ \mathbf{P}_{[k]}^- &= \mathbf{A}_{[k]} \mathbf{P}_{[k-1]} \mathbf{A}_{[k]}^T + \mathbf{W}_{[k]} \mathbf{Q}_{[k-1]} \mathbf{W}_{[k]}^T. \end{aligned} \quad (5.15)$$

The equations in (5.15) predict the state $\mathbf{x}_{[k]}$ and the error covariance $\mathbf{P}_{[k]}$ at time step k based on the results from the previous time step $k-1$.

Then, the EKF measurement update equations are used to correct these predictions based on the actual measurements at time step k , compare (5.16).

$$\begin{aligned} \mathbf{K}_{[k]} &= \mathbf{P}_{[k]}^- \mathbf{H}_{i,[k]}^T \left(\mathbf{H}_{i,[k]} \mathbf{P}_{[k]}^- \mathbf{H}_{i,[k]}^T + \mathbf{V}_{i,[k]} \mathbf{R}_{[k]} \mathbf{V}_{i,[k]}^T \right)^{-1} \\ \hat{\mathbf{x}}_{[k]} &= \hat{\mathbf{x}}_{[k]}^- + \mathbf{K}_{[k]} \left(\mathbf{y}_{i,[k]} - \mathbf{h}_i(\hat{\mathbf{x}}_{[k]}^-, \mathbf{0}) \right) \\ \mathbf{P}_{[k]} &= (\mathbf{I}_{25 \times 25} - \mathbf{K}_{[k]} \mathbf{H}_{i,[k]}) \mathbf{P}_{[k]}^-. \end{aligned} \quad (5.16)$$

The first equation computes the Kalman gain $\mathbf{K}_{[k]}$, the second equation updates the estimate $\hat{\mathbf{x}}_{[k]}$ with the current measurement $\mathbf{y}_{i,[k]}$ and the third equation updates the error covariance $\mathbf{P}_{[k]}$.

Unscented Kalman Filter (UKF). As detailed in the previous paragraph, the EKF approximates the state distribution by a Gaussian random variable, which is then propagated through the first-order linearization of the nonlinear system. This can introduce large errors in the mean and covariance, particularly if the system is highly nonlinear on the considered time scale.

The UKF is based on the intuition that, when having a fixed number of parameters, it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function [66]. Hence, instead of a linearization, the UKF uses a set of deterministic sample points (sigma points) to parameterize the mean and the covariance. The state distribution is still approximated by a Gaussian random variable, but is now represented through the set of sigma points. The sample points capture the true mean and covariance of the Gaussian random variable. The sigma points are then propagated through the actual nonlinear system (and not through a linearization). They capture the posterior mean and covariance accurately to the third order whereas with the EKF, only first order accuracy is achieved [141]. In addition to the higher accuracy, the UKF can be applied to a broader class of system since it does not require the calculation of partial derivatives. In contrast to particle filters, the UKF selects the sample points in a deterministic way. Hence, compared to e.g. Monte Carlo methods, less sample points are needed. The UKF belongs to the class of so called Sigma Point Kalman Filters (SPKF), together with Central Difference Kalman filter (CDKF), Square-Root SPKF and SPKF smoother [141]. In the following, the basic algorithm of the UKF is summarized - for the detailed idea and implementation notes of the filter, see [64, 65].

For a state vector \mathbf{x} of dimension L , the $2L + 1$ sigma points (vectors) $\boldsymbol{\chi}_i$ and their corresponding weights for the covariance $\mathbf{G}_i^{(c)}$ and the mean $\mathbf{G}_i^{(m)}$ are created according to the following procedure

$$\begin{aligned} \boldsymbol{\chi}_0 &= \mathbf{x}_{[k]} \\ \boldsymbol{\chi}_i &= \begin{cases} \mathbf{x}_{[k]} + (\sqrt{(L + \lambda)\mathbf{P}_{[k]}})_i & \text{for } i = 1, \dots, L \\ \mathbf{x}_{[k]} - (\sqrt{(L + \lambda)\mathbf{P}_{[k]}})_{i-L} & \text{for } i = L + 1, \dots, 2L \end{cases} \\ \mathbf{G}_0^{(c)} &= \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) & \mathbf{G}_0^{(m)} &= \frac{\lambda}{L + \lambda} \\ \mathbf{G}_i^{(c)} &= \mathbf{G}_i^{(m)} = \frac{1}{2(L + \lambda)} & \text{for } i = 1, \dots, 2L \end{aligned} \quad (5.17)$$

$$\begin{aligned} \text{with } \lambda &= \alpha^2(L + \kappa) - L \\ L &= \dim(\mathbf{x}) \end{aligned}$$

where $\mathbf{P}_{[k]}$ is the covariance matrix of $\mathbf{x}_{[k]}$ and $(\sqrt{(L + \lambda)\mathbf{P}_{[k]}})_i$ the i -th column of the matrix square root. The parameters α and κ define the distribution of the sigma points around the mean and are typically set to small positive values. The parameter β is used to include prior knowledge on the distribution of $\mathbf{x}_{[k]}$. For Gaussian distributions, the optimal choice is $\beta = 2$ [141]. For the UKF-based observers in this work, the parameters are set to $\alpha = 10^{-3}$, $\kappa = 0$, and $\beta = 2$.

With the sigma points and the corresponding weights defined in (5.17), the basic for-

mulation of the UKF is as follows: in the first step, sigma points for the augmented state vector $\mathbf{x}_{[k-1]}^a = [\mathbf{x}_{[k-1]} \quad \mathbf{w}_{[k-1]} \quad \boldsymbol{\nu}_{[k-1]}]^T$ and the corresponding augmented covariance matrix $\mathbf{P}_{[k-1]}^a = \text{diag}([\mathbf{P}_{[k-1]} \quad \mathbf{Q} \quad \mathbf{R}])$ are calculated, leading to

$$\boldsymbol{\chi}_{i,[k-1]}^a = \left[\boldsymbol{\chi}_{i,[k-1]}^x \quad \boldsymbol{\chi}_{i,[k-1]}^w \quad \boldsymbol{\chi}_{i,[k-1]}^\nu \right]^T. \quad (5.18)$$

Next, the sigma points are propagated through process and measurement model

$$\boldsymbol{\chi}_{i,[k]}^x = \mathbf{f}(\boldsymbol{\chi}_{i,[k-1]}^x, \boldsymbol{\chi}_{i,[k-1]}^w) \quad \boldsymbol{\mathcal{Y}}_{i,[k]} = \mathbf{h}(\boldsymbol{\chi}_{i,[k-1]}^x, \boldsymbol{\chi}_{i,[k-1]}^\nu) \quad \text{for } i = 0, \dots, 2L. \quad (5.19)$$

The time update equations (prediction) are then given by

$$\begin{aligned} \hat{\mathbf{x}}_{[k]}^- &= \sum_{i=0}^{2L} \mathbf{G}_i^{(m)} \boldsymbol{\chi}_{i,[k]}^x \\ \mathbf{P}_{[k]}^- &= \sum_{i=0}^{2L} \mathbf{G}_i^{(c)} \left[\boldsymbol{\chi}_{i,[k]}^x - \hat{\mathbf{x}}_{[k]}^- \right] \left[\boldsymbol{\chi}_{i,[k]}^x - \hat{\mathbf{x}}_{[k]}^- \right]^T \\ \hat{\boldsymbol{\mathcal{Y}}}_{[k]}^- &= \sum_{i=0}^{2L} \mathbf{G}_i^{(m)} \boldsymbol{\mathcal{Y}}_{i,[k]}. \end{aligned} \quad (5.20)$$

The measurement update equations (correction step) are

$$\begin{aligned} \mathbf{K}_{[k]} &= \mathbf{P}_{xy,[k]} \mathbf{P}_{yy,[k]}^{-1} \\ \hat{\mathbf{x}}_{[k]} &= \hat{\mathbf{x}}_{[k]}^- + \mathbf{K}_{[k]} \left(\boldsymbol{\mathcal{Y}}_{i,[k]} - \hat{\boldsymbol{\mathcal{Y}}}_{[k]}^- \right) \\ \mathbf{P}_{[k]} &= \mathbf{P}_{[k]}^- - \mathbf{K}_{[k]} \mathbf{P}_{yy,[k]} \mathbf{K}_{[k]}^T, \end{aligned} \quad (5.21)$$

with

$$\begin{aligned} \mathbf{P}_{yy,[k]} &= \sum_{i=0}^{2L} \mathbf{G}_i^{(c)} \left[\boldsymbol{\mathcal{Y}}_{i,[k]} - \hat{\boldsymbol{\mathcal{Y}}}_{[k]}^- \right] \left[\boldsymbol{\mathcal{Y}}_{i,[k]} - \hat{\boldsymbol{\mathcal{Y}}}_{[k]}^- \right]^T \\ \mathbf{P}_{xy,[k]} &= \sum_{i=0}^{2L} \mathbf{G}_i^{(c)} \left[\boldsymbol{\chi}_{i,[k]}^x - \hat{\mathbf{x}}_{[k]}^- \right] \left[\boldsymbol{\mathcal{Y}}_{i,[k]} - \hat{\boldsymbol{\mathcal{Y}}}_{[k]}^- \right]^T. \end{aligned}$$

In contrast to the EKF, the UKF does not require analytical derivatives, compare (5.14). The computational complexity of EKF and UKF is comparable [141].

5.2.4 Simulation

The purpose of the following simulation study is the evaluation of the four observer designs (EKF with \mathbf{y}_1 , EKF with \mathbf{y}_2 , UKF with \mathbf{y}_1 , UKF with \mathbf{y}_2). Particularly, the simulation of the system facilitates the investigation of the observer performance for changing environment conditions. In the following, this will be denoted as scenario I.

While gravity has a significant effect on the measured forces and torques, the compensation is relatively easy as its influence is only configuration-dependent. In contrast, the compensation of inertial effects is more challenging since these forces/torques also depend

on acceleration and (rotational) velocity. Hence, in order to facilitate the presentation and discussion of the results, zero gravity was assumed in the simulation.

Scenario. The sampling time of the filter is $T_s = 1$ ms. The tool mass is $m = 2$ kg and the inertia tensor $\mathbf{J}_C^C = \text{diag}([3, 2, 1])$ kg·m². The vector from the center of mass to the force sensor coordinate frame is $\mathbf{r}_{CS}^C = [0.1 \text{ m}, 0.1 \text{ m}, 0.1 \text{ m}]^T$. In the simulation, the contact point of the tool with the environment coincides with the center of gravity. The process noise covariance is set to

$$\begin{aligned} \mathbf{Q} &= \text{diag}([\mathbf{Q}_p, \mathbf{Q}_o, \mathbf{Q}_v, \mathbf{Q}_\omega, \mathbf{Q}_{F_S}, \mathbf{Q}_{F_E}, \mathbf{Q}_{M_S}, \mathbf{Q}_{M_E}]) \\ \text{with } \mathbf{Q}_p &= 0.5T_s^4/m^2 \mathbf{I}_{3 \times 3} & \mathbf{Q}_o &= 0.5T_s^4 \mathbf{J}_C^{-1} \mathbf{J}_C^{-1} \mathbf{I}_{3 \times 3} \\ \mathbf{Q}_v &= T_s^2/m^2 \mathbf{I}_{3 \times 3} & \mathbf{Q}_\omega &= T_s^2 \mathbf{J}_C^{-1} \mathbf{J}_C^{-1} \mathbf{I}_{3 \times 3} \\ \mathbf{Q}_{F_S} &= \mathbf{Q}_{M_S} = \mathbf{Q}_{F_E} = \mathbf{Q}_{M_E} = 100T_s^2 \mathbf{I}_{3 \times 3}. \end{aligned} \quad (5.22)$$

The actual measurement noise covariances are

$$\begin{aligned} \mathbf{R}_1 &= \text{diag}([\mathbf{R}_p, \mathbf{R}_o, \mathbf{R}_{F_S}, \mathbf{R}_{M_S}]) \\ \mathbf{R}_2 &= \text{diag}([\mathbf{R}_p, \mathbf{R}_o, \mathbf{R}_{F_S}, \mathbf{R}_{M_S}, \mathbf{R}_a, \mathbf{R}_\alpha, \mathbf{R}_\omega,]) \end{aligned} \quad (5.23)$$

with $\mathbf{R}_p = 10^{-6}$ m², $\mathbf{R}_o = 10^{-6}$, $\mathbf{R}_{F_S} = 10^{-3}$ N², $\mathbf{R}_{M_S} = 10^{-3}$ Nm², $\mathbf{R}_a = 10^{-2}$ m²/s⁴, $\mathbf{R}_\alpha = 10^{-2}$ rad²/s⁴, and $\mathbf{R}_\omega = 10^{-3}$ rad²/s². In the simulation, it is assumed that process and measurement covariances are constant. Then, the performance of the filter can be optimized by taking sensor measurements and determining the values of \mathbf{R}_i off-line. In reality, however, it is likely that the measurement noise varies over time. To take that effect into account, three different cases are considered in the filter evaluation:

$$\text{A) } \bar{\mathbf{R}}_i = \mathbf{R}_i \quad \text{B) } \bar{\mathbf{R}}_i = 10^2 \mathbf{R}_i \quad \text{C) } \bar{\mathbf{R}}_i = 10^{-2} \mathbf{R}_i \quad (5.24)$$

with $i \in \{1, 2\}$. While case A) reflects a perfect knowledge of the actual measurement noise level, the cases B) and C) simulate an over- and an underestimation of the actual measurement noise level. Hence, the filter performance in the latter two cases is a measure for the robustness of the filter towards changing operating conditions.

In the simulation scenario I, the tool executes a sinusoidal motion in x -direction combined with a rotation of $-\pi$ around the x -axis, see Fig. 5.4. The measured translational and rotational tool accelerations are depicted in Fig. 5.5. During this motion, the following contact forces and torques occur:

$$\begin{aligned} \mathbf{F}_E &= \begin{cases} [20 \text{ N}, 0 \text{ N}, 0 \text{ N}]^T, & \text{if } 1 \text{ s} < t < 4 \text{ s} \\ [0 \text{ N}, 0 \text{ N}, 0 \text{ N}]^T, & \text{else.} \end{cases} \\ \mathbf{M}_E &= \begin{cases} [0 \text{ Nm}, 0 \text{ Nm}, 5 \text{ Nm}]^T, & \text{if } 2 \text{ s} < t < 3 \text{ s} \\ [0 \text{ Nm}, 0 \text{ Nm}, 0 \text{ Nm}]^T, & \text{else.} \end{cases} \end{aligned} \quad (5.25)$$

To evaluate the performance of the four filter designs, the force and torque estimation

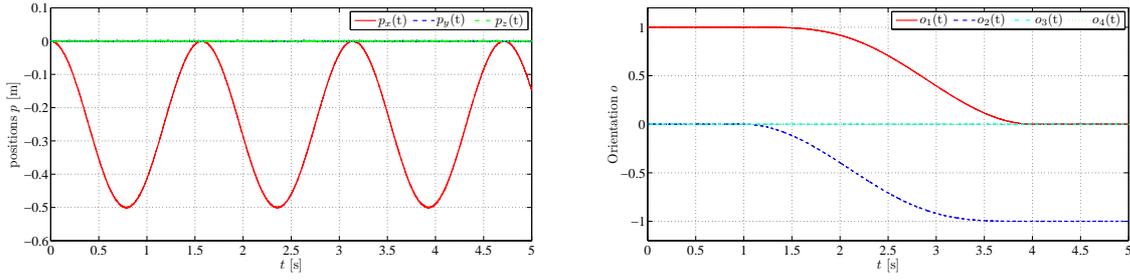


Fig. 5.4: Scenario I: Measured tool position (left) and orientation (right)

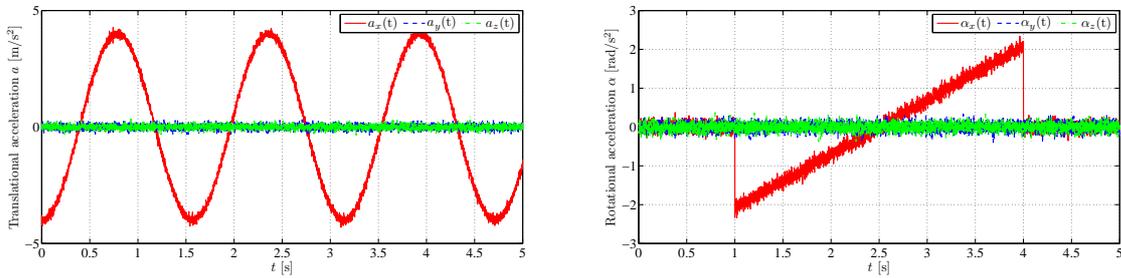


Fig. 5.5: Scenario I: Measured translational (left) and rotational tool acceleration (right)

errors at time step $[k]$ are defined as

$$\begin{aligned} \mathbf{e}_{F[k]} &= \mathbf{F}_{E[k]} - \hat{\mathbf{F}}_{E[k]} \\ \mathbf{e}_{M[k]} &= \mathbf{M}_{E[k]} - \hat{\mathbf{M}}_{E[k]}. \end{aligned} \quad (5.26)$$

For the evaluation, the performance of the filter with respect to both continuous and discrete changes of the environment forces and/or torques is of interest. Hence, the simulation is separated into different segments and each time step $[k]$ is associated either with phase c (continuous) or phase d (discrete) according to

$$\begin{aligned} b_{c,[k]} &= \begin{cases} 1, & \text{if } t[k] \in t_c \\ 0, & \text{else} \end{cases} \\ b_{d,[k]} &= 1 - b_{c,[k]}. \end{aligned} \quad (5.27)$$

Discrete changes of the environment forces and torques occur at $t = 1$ s, $t = 2$ s, $t = 3$ s, and $t = 4$ s, compare Eq. (5.25). Hence, the continuous phase is defined as

$$t_c = \{[0, 1) \vee [1.5, 2) \vee [2.5, 3) \vee [3.5, 4) \vee [4.5, 5]\}. \quad (5.28)$$

With (5.26)-(5.28), the scalar error measures

$$\begin{aligned}
 e_{F,c} &= \sum_{k=1}^N T_s b_{c,[k]} \|\mathbf{e}_{F[k]}\| & e_{F,d} &= \sum_{k=1}^N T_s b_{d,[k]} \|\mathbf{e}_{F[k]}\| \\
 e_{M,c} &= \sum_{k=1}^N T_s b_{c,[k]} \|\mathbf{e}_{M[k]}\| & e_{M,d} &= \sum_{k=1}^N T_s b_{d,[k]} \|\mathbf{e}_{M[k]}\|
 \end{aligned} \tag{5.29}$$

are defined. In addition, the error measures

$$e_{F,\Sigma} = e_{F,c} + e_{F,d} \quad \text{and} \quad e_{M,\Sigma} = e_{M,c} + e_{M,d} \tag{5.30}$$

are used to evaluate the combined performance in phases c and d .

Results. For case A), the measured sensor force, the true environment force, and the estimated environment force (UKF with \mathbf{y}_2) in x -direction are depicted in Fig. 5.6(a). The force estimation errors for the four different filter designs are illustrated in Fig. 5.6(c). The forces along y - and z -axis are not displayed since no environment forces were applied in this direction. But they are, of course, considered in the calculation of the error measures. The measured sensor torques, the true environment torques, and the estimated environment torques (UKF with \mathbf{y}_2) around the three axis are depicted in Fig. 5.6(b), 5.6(e), and 5.6(f). The corresponding estimation errors for the four filters are shown in Fig. 5.6(d), 5.6(g), and 5.6(h). It should be noted that all three measured torques show significant deviations from zero, despite the fact that an environment torque is only applied around the z -axis (from $t = 2$ s to $t = 3$ s). These torques are created by two sources: the environment force $F_{E,x}$ and the inertial force $F_{I,x}$ due to the translational acceleration along the x -axis. The latter force also leads to measured torques since the tools center of mass and the force sensor coordinate system do not coincide.

Discussion. The quantitative results of the simulation for the cases A) - C) are summarized in Tab.5.1 using the six error measures stated in (5.29) and (5.30). As will be detailed in the following, these results provide interesting information on the performance on the different filter designs. The first line of the table shows the values of the error measures when the measured forces \mathbf{F}_S and torques \mathbf{M}_S are assumed to be identical with the environment forces and torques. This can be considered as the worst case scenario. As \mathbf{F}_S and \mathbf{M}_S are unfiltered data, the results are in this case independent of the assumed measurement noise. Hence, the error measures are identical for the cases A) to C).

In case A), when using an observer design based on \mathbf{y}_1 (pose and F/T measurements), the combined error measures $e_{F,\Sigma}$ ($e_{M,\Sigma}$) can be reduced by appr. 82% (90%) compared to the worst case scenario. Here, EKF and UKF design are performing almost equally well. By adding an acceleration sensor and using measurements \mathbf{y}_2 fused from the three sensor types, the performance can be further improved: the combined error measure $e_{F,\Sigma}$ is reduced by 63.9 % for the EKF and by 65.2 % for the UKF (compared to the filter with \mathbf{y}_1). It should be noted that the additional acceleration measurements mainly improve the error measure $e_{F,d}$ which evaluates discrete changes, whereas the improvements of $e_{F,c}$

are of minor influence. The same holds for the combined error measure $e_{M,\Sigma}$ when a filter design with \mathbf{y}_2 instead of \mathbf{y}_1 is used: The measure is reduced by 71.4% for the EKF and by 72.8% for the UKF. Again, the improvements are mainly associated with $e_{M,d}$. This is not surprising since the acceleration measurements are especially beneficial when sudden changes in the measurements occur.

In case B), the performance of all four filter designs degrades compared to case A). This is not surprising since the actual measurement noise is overestimated by factor 10^2 . However, all filters still show a significant improvement compared to the raw measurements: the error measure $e_{F,\Sigma}$ is improved between 60.0% (EKF with \mathbf{y}_1) and 65.9% (UKF with \mathbf{y}_2), the error measure $e_{M,\Sigma}$ between 80.7% (EKF with \mathbf{y}_1) and 89.5% (UKF with \mathbf{y}_2). Remarkably, in case B, the error measure $e_{F,c}$ of the observer designs with \mathbf{y}_1 is smaller than that of their counterparts with \mathbf{y}_2 . However, considering the combined measure $e_{F,\Sigma}$, the designs with \mathbf{y}_1 are still superior.

In case C), the underestimation of the measurement noise by factor 10^{-2} results for all four filters in a performance which is between their respective performance in cases A) and B). However, the results of the filter designs with \mathbf{y}_1 are only slightly better than in case B): For the EKF (UKF), the measure $e_{F,\Sigma}$ is improved by 11.7% (1.8%) and the measure $e_{M,\Sigma}$ by 10.3% (10.5%). In contrast, the results of the filter designs with \mathbf{y}_1 are in case C) significantly better than in case B): for the EKF (UKF), the measure $e_{F,\Sigma}$ is improved by 73.8% (72.5%) and the measure $e_{M,\Sigma}$ by 49.8% (47.8%). Also, for the filter designs with \mathbf{y}_2 , it is interesting to see that the decline in performance compared to case A) is mainly caused by a worse performance in the continuous phases (error measures $e_{F,c}$ and $e_{M,c}$). The error measures $e_{F,d}$ and $e_{M,d}$ for discrete changes, however, are hardly affected by the underestimation of the measurement noise. This result corresponds with the intuitive idea of the functional principles of the filters.

Overall, the UKF based design with \mathbf{y}_2 showed the best performance although in most cases, there is only a small improvement when compared to the EKF with \mathbf{y}_2 . As an additional advantage of the design with the UKF, the calculation of the partial derivatives of process and measurement equation is not necessary, compare (5.14). While there is a significant performance difference between the designs with \mathbf{y}_2 and \mathbf{y}_1 , the observers based on the latter measurement vector still show a fundamental improvement compared to the raw measurements. For the designs with \mathbf{y}_1 , the performance of EKF and UKF is comparable with each of them giving better results in some cases.

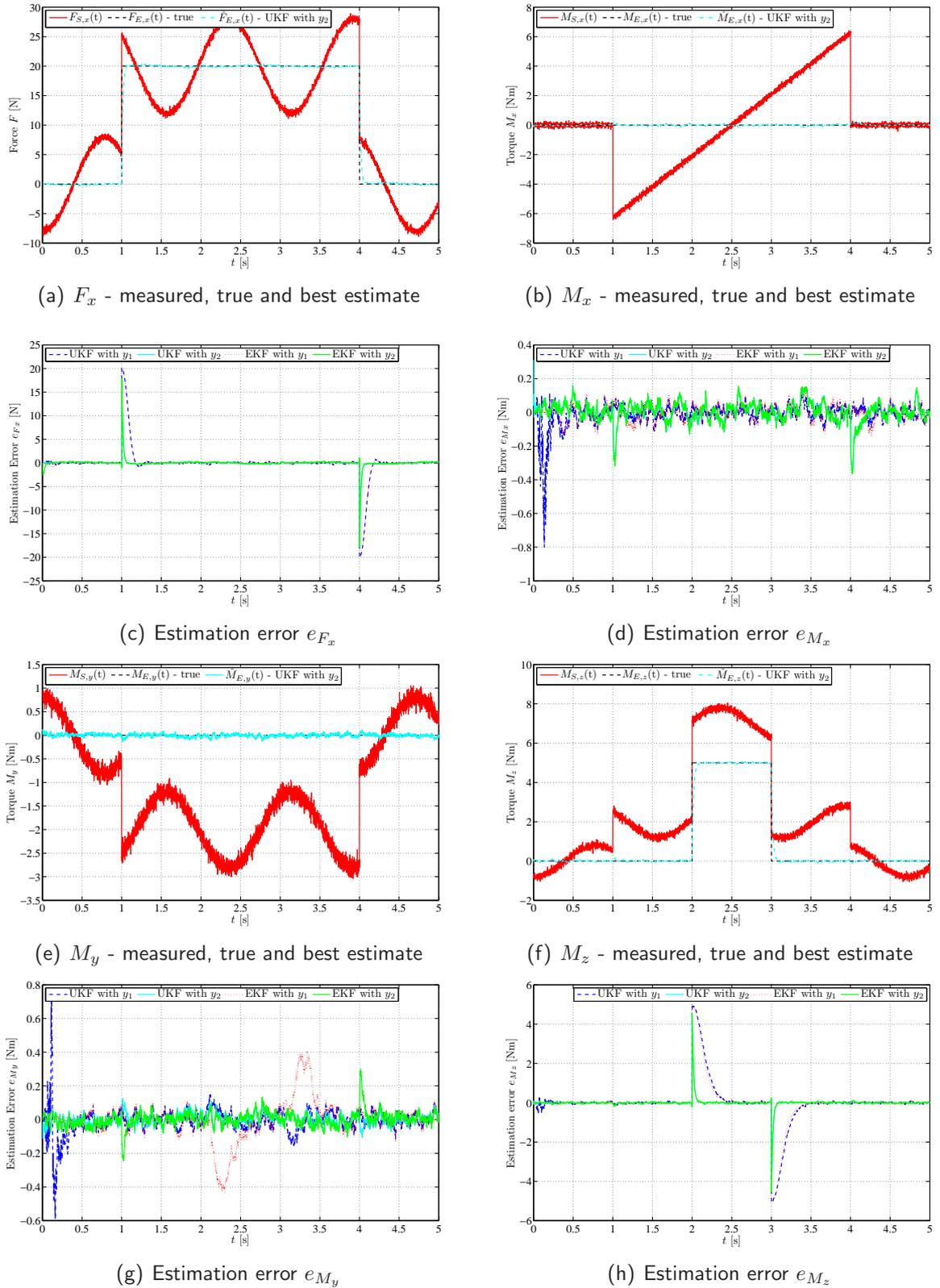


Fig. 5.6: Scenario I - Case A): (a) measured force $F_{S,x}$, true environment force $F_{E,x}$ and best estimate of the environment force $\hat{F}_{E,x}$, (c) estimation error e_{F_x} for the two UKF and the two EKF filter designs. The figure pairs (b) & (d), (e) & (g), and (f) & (h) show the corresponding results for the torques around the x , y , and z -axis.

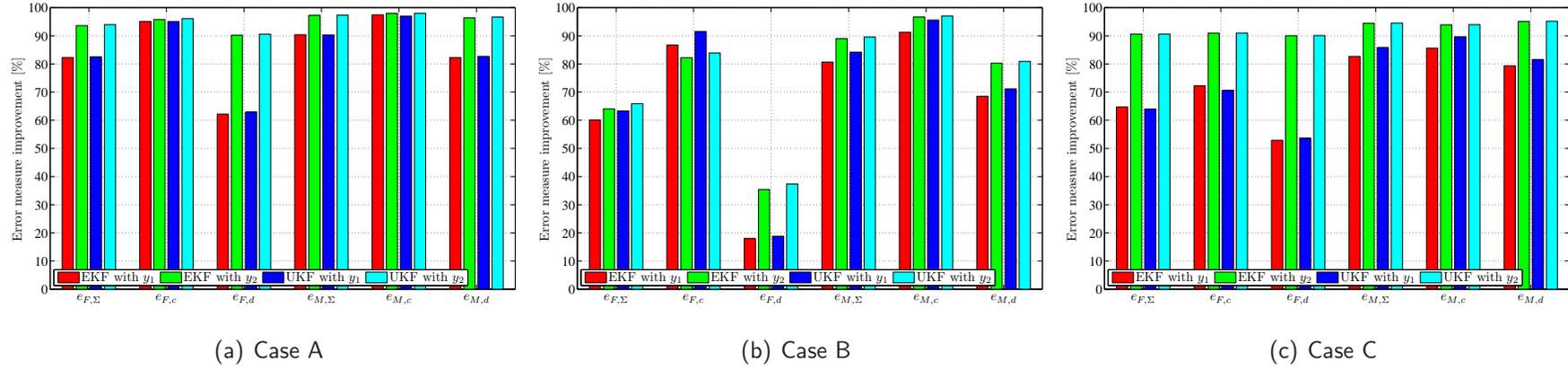
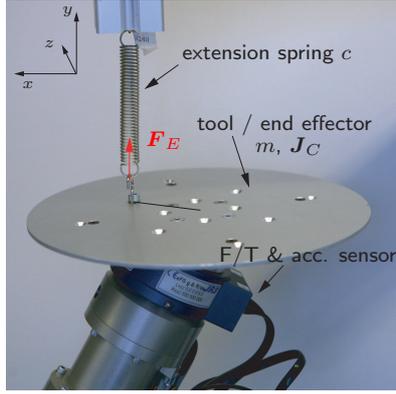


Fig. 5.7: Error measures: Improvements of the four filter designs compared to the measurements F_S , M_S .

	Case A						Case B						Case C					
	$e_{F,\Sigma}$	$e_{F,c}$	$e_{F,d}$	$e_{M,\Sigma}$	$e_{M,c}$	$e_{M,d}$	$e_{F,\Sigma}$	$e_{F,c}$	$e_{F,d}$	$e_{M,\Sigma}$	$e_{M,c}$	$e_{M,d}$	$e_{F,\Sigma}$	$e_{F,c}$	$e_{F,d}$	$e_{M,\Sigma}$	$e_{M,c}$	$e_{M,d}$
meas. F_S, M_S	28.37	17.37	11.00	23.57	12.60	10.97	28.37	17.37	11.00	23.57	12.60	10.97	28.37	17.37	11.00	23.57	12.60	10.97
EKF with y_1	5.02	0.86	4.16	2.27	0.32	1.94	11.33	2.31	9.03	4.55	1.10	3.46	10.01	4.83	5.18	4.08	1.81	2.27
UKF with y_1	4.95	0.87	4.08	2.28	0.38	1.90	10.41	1.48	8.93	3.72	0.56	3.17	10.22	5.11	5.11	3.33	1.30	2.03
EKF with y_2	1.81	0.73	1.08	0.65	0.25	0.40	10.20	3.09	7.11	2.59	0.42	2.17	2.67	1.57	1.10	1.30	0.77	0.54
UKF with y_2	1.72	0.68	1.04	0.62	0.25	0.37	9.68	2.79	6.89	2.47	0.37	2.10	2.66	1.56	1.09	1.29	0.76	0.53

Tab. 5.1: Scenario I: Error measures $e_{M,\Sigma}$, $e_{F,c}$, $e_{F,d}$, $e_{M,\Sigma}$, $e_{M,c}$, and $e_{M,d}$ for the different filter designs and cases A) - C). In each column, the filter with the best performance is highlighted.



Parameters	Units
$m = 0.98$	[kg]
$\mathbf{J}_C^C = 10^{-3} \begin{bmatrix} 1.83 & 0.161 & 0.414 \\ 0.161 & 1.87 & -0.436 \\ 0.414 & -0.436 & 2.82 \end{bmatrix}$	[kg·m ²]
$\mathbf{r}_{CS}^C = [0.0001, 0.00001, -0.0446]^T$	[m]
$\mathbf{r}_{CE}^C = [0.0549, -0.015, 0.0096]^T$	[m]
$T_s = 0.001$	[s]
$c = 0.303$	[N/mm]

Fig. 5.8: Scenario II: Experimental setup with a spring attached to one end effector of the dual-arm robot (left) and dynamic parameters of the tool, relative positions of the F/T sensor and the contact point of the environment interaction and the sampling time (right).

5.2.5 Experiments

In order to verify the simulation, the filter designs were also evaluated using real experimental data. In the following, this will be denoted as scenario II. As in simulation scenario I, the main focus is on the compensation of inertial effects and the rejection of measurement noise. While gravitational forces and torques occurred during the experiments, their effects have been eliminated from the measured forces and torques that are shown in the results and the discussion.

Scenario. The experimental setup and the parameters of the experiment are depicted in Fig. 5.8: an extension spring with spring constant $c = 0.303$ N/mm is attached to the end effector in order to produce a known environment force when the robot is moving. The dynamic parameters of the end effector were determined based on its CAD model. The forces, torques, and accelerations are measured with a twelve DOF sensor from JR3. The measurement noise covariances \mathbf{R}_i were determined with the robot control running and the robot commanded to remain in a stationary pose. The following values were obtained:

$$\begin{aligned} \mathbf{R}_1 &= \text{diag}([\mathbf{R}_p, \mathbf{R}_o, \mathbf{R}_{F_S}, \mathbf{R}_{M_S}]) \\ \mathbf{R}_2 &= \text{diag}([\mathbf{R}_p, \mathbf{R}_o, \mathbf{R}_{F_S}, \mathbf{R}_{M_S}, \mathbf{R}_a, \mathbf{R}_\alpha, \mathbf{R}_\omega,]) \end{aligned} \quad (5.31)$$

with $\mathbf{R}_p = 10^{-6}$ m², $\mathbf{R}_o = 10^{-6}$, $\mathbf{R}_{F_S} = 3 \cdot 10^{-3}$ N², $\mathbf{R}_{M_S} = 10^{-3}$ Nm², $\mathbf{R}_a = 10^{-3}$ m²/s⁴, $\mathbf{R}_\alpha = 1$ rad²/s⁴, and $\mathbf{R}_\omega = 10^{-3}$ rad²/s². The process noise covariance \mathbf{Q} is set to the same values as in the simulation, see (5.22). From time $t = 0$ s to 1 s, the end effector moves to a specified position \mathbf{p}_0 to attach the spring. Starting at $t_1 = 1$ s, the end effector executes a sinusoidal motion in y -direction

$$p_y(t) = p_{0,y} + 0.03 (1 - e^{-0.2(t-t_1)}) \sin(15(t - t_1)), \quad (5.32)$$

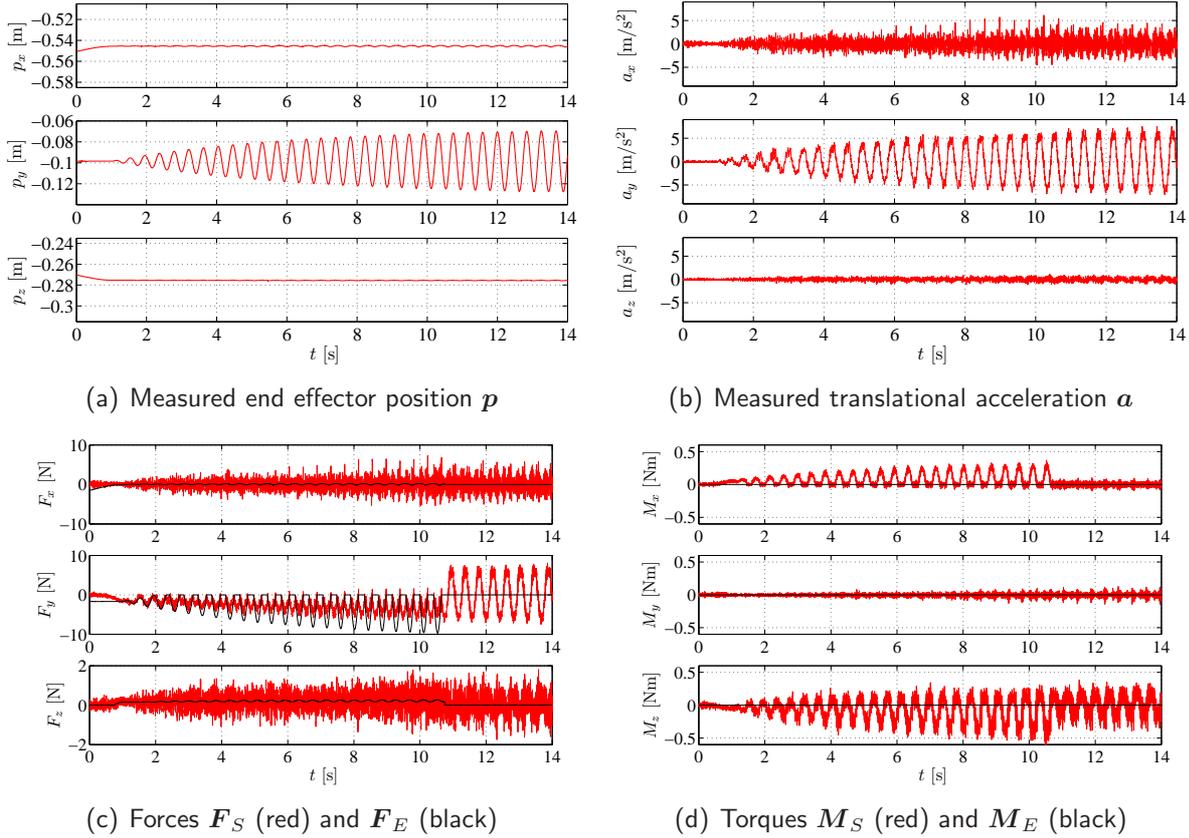


Fig. 5.9: Scenario II: (a) measured position p and (b) measured translational acceleration a of the end effector. Figures (c)/(d) show the measured forces F_S / torques M_S and the true environment forces F_E / torques M_E .

while $p_x(t)$ and $p_z(t)$ remain constant. After 23 cycles ($t = 10.7$ s), the spring is detached from the end effector while the sinusoidal motion is continued until the end of the experiment at $t = 14$ s.

The measured end effector position and acceleration are depicted in Fig. 5.9(a) and (b). The forces F_S measured with the F/T-sensor and the true environment forces F_E are illustrated in Fig. 5.9(c). The true environment forces caused by the spring are calculated based on the actual end effector position during the task execution. As the spring is aligned with the y -axis, a perfect trajectory execution would cause only forces in y -direction. However, the error between desired and actual position also results in small forces in z -direction. Due to inertial effects and noise, the measured sensor forces deviate strongly from the actual forces exchanged with the environment, compare Fig. 5.9(c).

Similarly, Fig. 5.9(d) shows the measured torques M_S and the true environment torques M_E . As interaction with the environment occurs only via the extension spring, the true environment torques are zero. Again, there are significant deviations between the measured and the true environment torques. In addition to inertial effects and measurement noise, these deviations are also caused by spring: the off-center contact point creates torques around the x - and z -axis.

For the evaluation, the force and torque estimation errors as defined in (5.26) are used. In contrast to the simulation, there is no differentiation between continuous and discrete

phases. Hence, only the combined error measures $e_{F,\Sigma}$ and $e_{M,\Sigma}$ are utilized for the evaluation of the performance, compare (5.30).

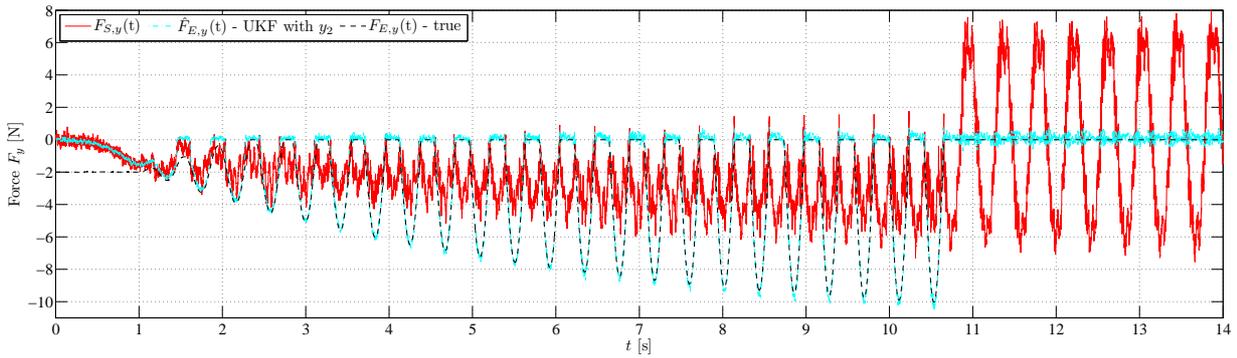
Results. The measured sensor force, the true environment force, and the estimated environment force (UKF with \mathbf{y}_2) in y -direction are depicted in Fig. 5.10(a). The force estimation errors for the four different filter designs are illustrated in Fig. 5.10(b). For the two observers based on measurement vector \mathbf{y}_1 , the estimation error shows very similar behavior: during contact with the environment, the estimates lag behind the actual forces leading to errors of up to 7 N. The estimation errors of the \mathbf{y}_2 -based designs are also almost identical. However, the maximum estimation error for these designs is only 1.8 N. From Fig. 5.10(a), it is evident that measured sensor forces can not be directly used in a force control scheme: in both phases, there is a significant deviation between the measurements and the actual environment forces. During the contact phase (from 1 s to 10.8 s), the environment/spring force \mathbf{F}_E partially compensates the inertial forces \mathbf{F}_I . During the free space motion (10.8 s to 14 s), the measured forces reflect the inertial forces caused by the sinusoidal motion with increasing amplitude.

The measured sensor torques, the true environment torques, and the estimated environment torques (UKF with \mathbf{y}_2) around the x - and z -axis are depicted in Fig. 5.12(a) and (c). The corresponding estimation errors for the four filters are shown in Fig. 5.12(b) and (d). As for the forces, the estimation errors for UKF and EKF with the same measurement vector are very similar. Furthermore, the maximum errors of the \mathbf{y}_2 -based designs (0.04 Nm) are considerably lower than those of the filters with \mathbf{y}_1 (0.1 Nm). In analogy to the forces, the deviations between measured torques \mathbf{M}_S and actual environment torques \mathbf{M}_E depicted in Fig. 5.12(a) and (c) illustrate the importance of the observer for a reliable force control performance.

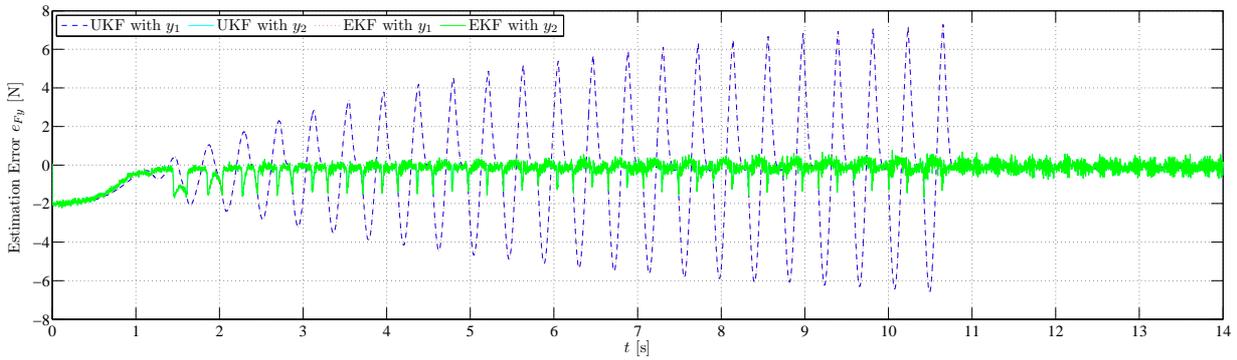
Discussion. The quantitative results are summarized in Fig. 5.11 using the two error measures $e_{F,\Sigma}$ and $e_{M,\Sigma}$. Again, the first row of the table shows the values of the error measures when the measured forces \mathbf{F}_S and torques \mathbf{M}_S are assumed to be identical with the environment forces and torques. This can be considered as the worst case scenario.

When using an observer design based on \mathbf{y}_1 (pose and F/T measurements), the combined error measures $e_{F,\Sigma}$ ($e_{M,\Sigma}$) can be reduced by appr. 57% (80%) compared to the worst case scenario. Here, EKF and UKF design are performing almost equally well. By adding an acceleration sensor and using measurements \mathbf{y}_2 fused from the three sensor types, the performance can be further improved: The combined error measure $e_{F,\Sigma}$ is reduced by 74.7 % for the EKF and by 74.6 % for the UKF (compared to the filter with \mathbf{y}_1). The same holds for the combined error measure $e_{M,\Sigma}$ when a filter design with \mathbf{y}_2 instead of \mathbf{y}_1 is used: The measure is reduced by 61.3% for the EKF and by 60.5% for the UKF.

Overall, the experimental results are in accordance with those of the simulation: First, EKF and UKF based observer designs perform almost equally well. Second, the designs with the extended measurement vector \mathbf{y}_2 show a significant performance improvement compared to designs based on \mathbf{y}_1 . The observers with \mathbf{y}_1 , in turn, still show significant improvements compared to the measured values.



(a) F_y - measured, true and best estimate



(b) Estimation error e_{F_y}

Fig. 5.10: Scenario II: (a) measured force $F_{S,y}$, true environment force $F_{E,y}$ and best estimate of the environment force $\hat{F}_{E,y}$, (b) estimation error e_{F_y} for the two UKF and the two EKF filter designs.

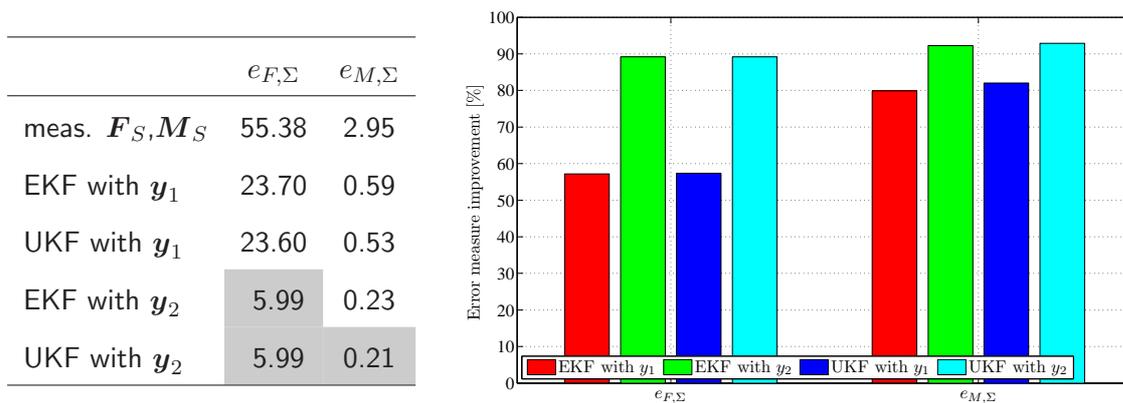


Fig. 5.11: Scenario II: Error measures - improvements of the four filter designs compared to the measurements F_S, M_S . In each column, the filters with the best performance are highlighted.

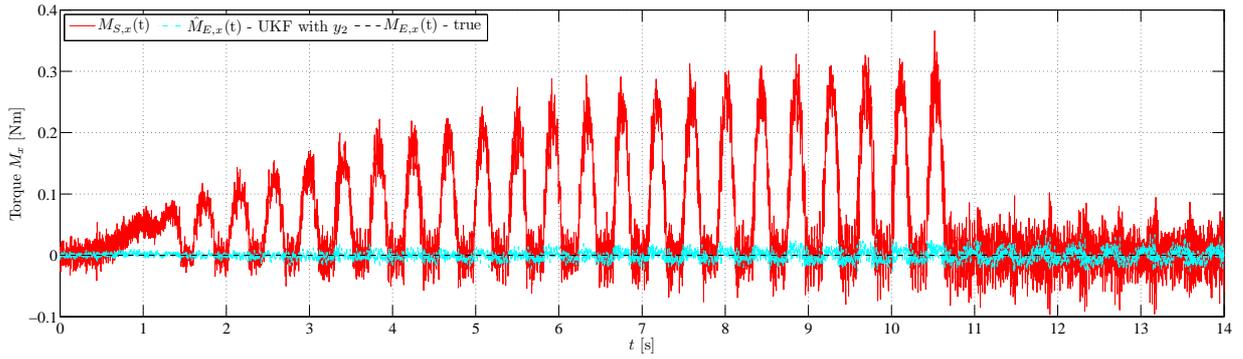
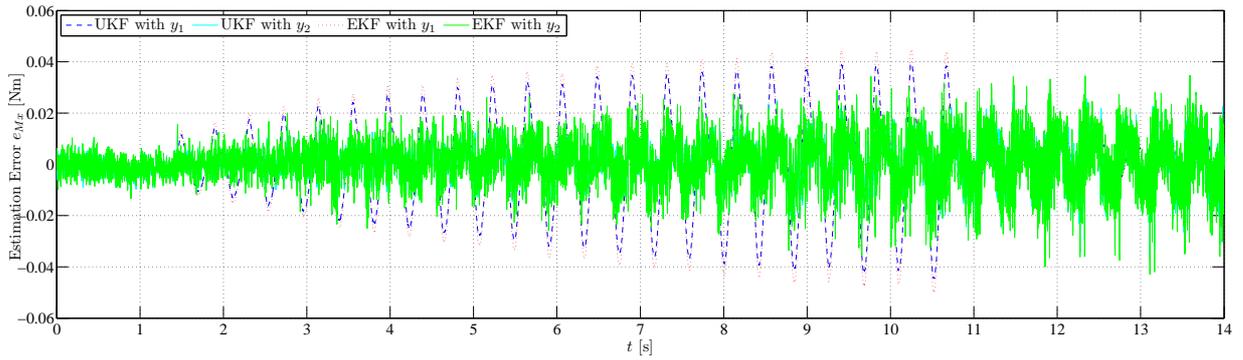
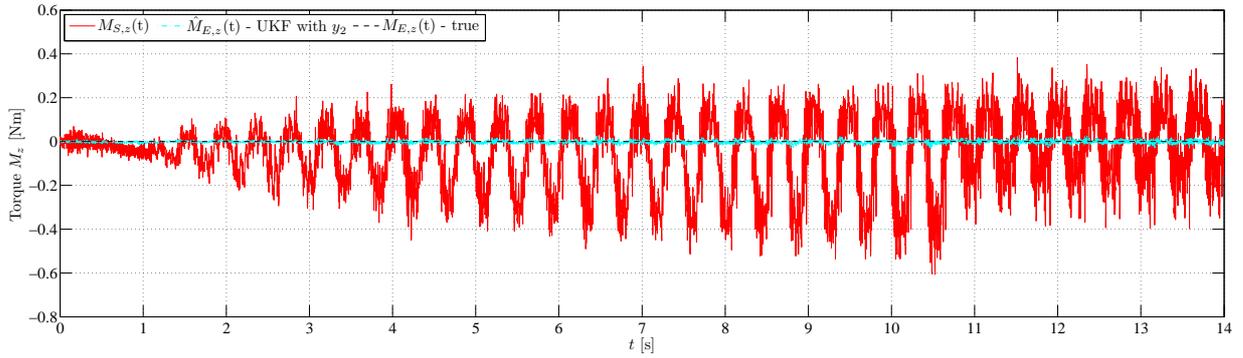
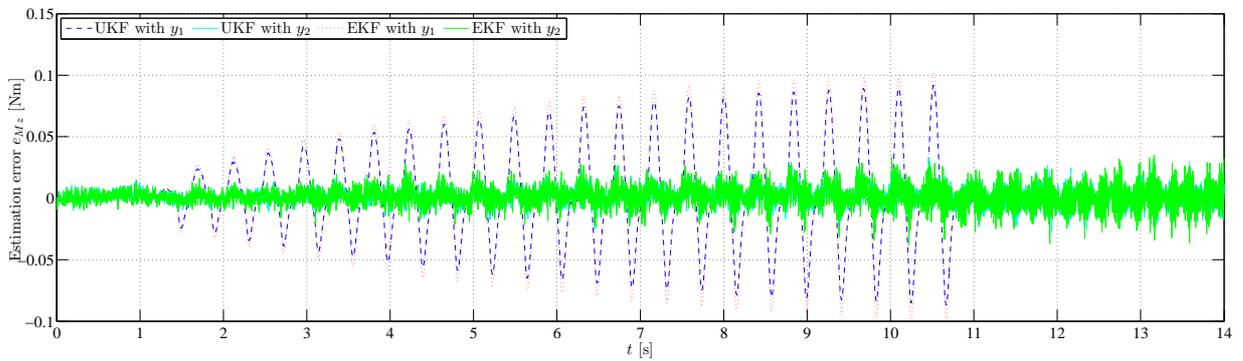
(a) M_x - measured, true and best estimate(b) Estimation error e_{M_x} (c) M_z - measured, true and best estimate(d) Estimation error e_{M_z}

Fig. 5.12: Scenario II: (a) measured force $M_{S,x}$, true environment force $M_{E,x}$ and best estimate of the environment force $\hat{M}_{E,x}$, (b) estimation error e_{M_x} for the two UKF and the two EKF filter designs. The figure pair (c) & (d) shows the corresponding results for the torque around the z -axis.

5.3 Motion and Interaction Control

The following section summarizes the robot control strategies that have been utilized in this thesis. Here, two different tasks have to be considered: first, the control of the robot during free space motions. Second, the interaction control of the robot when the manipulator is in contact with the environment. The former is realized with the inverse dynamics approach, the latter with a parallel force/position control scheme. In both cases, the task space formalism is used.

Commonly, interaction control schemes assume that the weight and the inertia of the tool are negligible [125]. Since this assumption does not hold for dynamic manipulation tasks, it is necessary to modify the interaction control scheme with the dynamic contact force/torque observer that has been presented in the previous section.

A detailed discussion on interaction control schemes can be found in [94]. For the dual-arm robot used in this work, impedance control schemes were evaluated by Stanczyk [132].

5.3.1 Task Space Motion Control

In general, the pose of the end effector is defined by

$$\mathbf{x}_M = [\mathbf{p}_M^T \quad \mathbf{o}_M^T]^T, \quad (5.33)$$

where $\mathbf{p}_M \in \mathbb{R}^3$ denotes the position and \mathbf{o}_M the orientation of the end effector. In *operational* space control, the orientation is described with three Euler angles. These are the minimal set of variables to represent orientation, resulting in $\mathbf{x}_M \in \mathbb{R}^6$. However, such an approach has two drawbacks: first, Euler angles have representation singularities. Second, they are inconsistent with the task geometry, see Natale for a detailed discussion [94]. To overcome these drawbacks, *task* space control schemes have been proposed, see e.g. Caccavale et al. [26]. Here, the orientation is represented using a non-minimal set of variables based on an angle-axis description. The most common angle-axis notation are unit quaternions, which are also used in this work. Consequently, the pose of the end effector is given by $\mathbf{x}_M = [\mathbf{p}_M^T \quad \mathbf{o}_e^T]^T \in \mathbb{R}^7$ with $\mathbf{o}_M = \{\eta, \boldsymbol{\epsilon}\}$. Details on the quaternion notation are given in App. A.3. When referring to the end effector pose \mathbf{x}_M , the following secondary subscripts are used: *d* (desired), *a* (actual), *c* (compliant), *r* (reference).

The dynamic model of a robot manipulator is given by

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{T}_E \quad \in \mathbb{R}^n \quad (5.34)$$

where $\mathbf{B}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ the coriolis and centrifugal torques, $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) / \mathbf{g}(\mathbf{q})$ the frictional / gravitational torques, $\boldsymbol{\tau}$ the joint driving torques, $\mathbf{J}(\mathbf{q})$ the manipulator jacobian and $\mathbf{T}_E = [\mathbf{F}_E^T \quad \mathbf{M}_E^T]^T$ the vector of environment forces/torques [126].

A block scheme of the motion control is shown in Fig. 5.13 as the gray part of the interaction control. It consists of three main modules: forward kinematics, pose control, and inverse dynamics.

Pose control. Given a (desired) reference trajectory ($\mathbf{x}_{M,r}$, $\dot{\mathbf{x}}_{M,r}$, and $\ddot{\mathbf{x}}_{M,r}$) and the actual robot pose and velocity ($\mathbf{x}_{M,a}$, $\dot{\mathbf{x}}_{M,a}$), the module determines the resolved acceleration \mathbf{a}_M in task space variables as

$$\mathbf{a}_M = \begin{bmatrix} \mathbf{a}_{M,p} \\ \mathbf{a}_{M,o} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{p}}_{M,r} + \mathbf{K}_{Dp}\Delta\dot{\mathbf{p}}_{M,ra} + \mathbf{K}_{Pp}\Delta\mathbf{p}_{M,ra} \\ \dot{\boldsymbol{\omega}}_{M,r} + \mathbf{K}_{Do}\Delta\boldsymbol{\omega}_{M,ra} + \mathbf{K}_{Po}\mathbf{R}_M\boldsymbol{\epsilon}_{M,ra} \end{bmatrix} \quad (5.35)$$

where $\Delta\dot{\mathbf{p}}_{M,ra}$ and $\Delta\boldsymbol{\omega}_{M,ra}$ are the velocity tracking errors, $\Delta\mathbf{p}_{M,ra}$ and $\boldsymbol{\epsilon}_{M,ra}$ the pose tracking errors, and \mathbf{K}_{Dp} , \mathbf{K}_{Pp} , \mathbf{K}_{Do} , and \mathbf{K}_{Po} are matrix gains as detailed in [94].

Inverse dynamics. The inverse dynamics module applies the feedback linearization technique to the robotic system in order to obtain a linear and decoupled system. To this end, the module calculates the joint torques that match the resolved acceleration. This is realized in two steps: first, the joint accelerations $\ddot{\mathbf{q}}$ that correspond to the resolved end effector acceleration have to be determined,

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \left(\mathbf{a}_M - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}}) \right). \quad (5.36)$$

In the second step, the equivalent joint torques are calculated based on the dynamic model given in (5.34).

Friction compensation. As part of the feedback linearization performed in the inverse dynamics module, a model for the joint friction is needed. Commonly, a viscous friction model is utilized because of its simplicity [94, 126]. While this is easy to implement, it captures only a part of the friction phenomena discussed in Sec. 2.5. Clearly, the accuracy of the friction compensator is highly dependent on the validity of the used model. For dynamic manipulation tasks, a large range of joint velocities occurs. Hence, it is necessary to adopt a friction model that is describing all the relevant effects. In this work, a static friction model is utilized that considers Coulomb, viscous, and static friction and also includes the Stribeck effect.

5.3.2 Interaction Control

Sec. 5.1 presented a brief overview on interaction control strategies. Comparing the existing concepts, the parallel approach developed by Chiaverini and Sciavicco is the best choice for the dynamic manipulation tasks considered in this work. This is due to two reasons: first, the method allows to specify the desired interaction forces/torques directly. Second, in contrast to hybrid force/position control, the controller structure does not need to be changed during different stages of the task execution [28].

The parallel approach, like all interaction control schemes, utilizes the measured sensor forces \mathbf{F}_S and torques \mathbf{M}_S either directly or after applying a low-pass filter [125]. This is reasonable, as in most tasks, the weight and the inertia of the tool are negligible. However, this assumption does not hold for dynamic robot motions where interaction forces occur during phases with significant acceleration of the end effector. Hence, it is necessary to

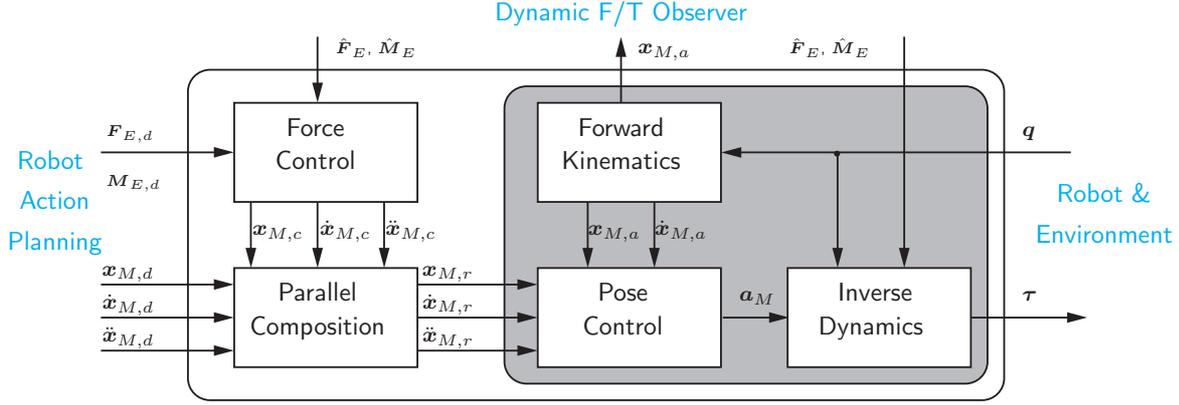


Fig. 5.13: Block diagram of the motion control (in gray) and the overall interaction control scheme based on parallel composition.

extend the force control scheme with one of the dynamic contact force/torque observer designs discussed in Sec. 5.2.

Fig. 5.13 shows the parallel approach for direct force control used in this work. The task space motion control is extended by adding two modules: force control and parallel composition. In the following, the two modules are briefly summarized. For a detailed discussion, see [28, 94]. In contrast to the original scheme, the control does not use the sensor forces/torques $\mathbf{F}_S/\mathbf{M}_S$ but the observed quantities $\hat{\mathbf{F}}_E/\hat{\mathbf{M}}_E$.

Force control. The module determines a *compliant* trajectory $\mathbf{x}_{M,c}$, $\dot{\mathbf{x}}_{M,c}$, and $\ddot{\mathbf{x}}_{M,c}$ based on the differential equations

$$\begin{aligned} \mathbf{K}_{AF}\ddot{\mathbf{p}}_{M,c} + \mathbf{K}_{VF}\dot{\mathbf{p}}_{M,c} &= \mathbf{F}_{E,d} - \mathbf{F}_E \\ \mathbf{K}_{AM}\dot{\boldsymbol{\omega}}_{M,c}^c + \mathbf{K}_{VM}\boldsymbol{\omega}_{M,c}^c &= \mathbf{M}_{E,d}^c - \mathbf{M}_{E}^c, \end{aligned} \quad (5.37)$$

where \mathbf{K}_{AF} , \mathbf{K}_{VF} , \mathbf{K}_{AM} , and \mathbf{K}_{VM} are matrix gains and the resulting compliant pose $\mathbf{x}_{M,c}$ incorporates an integral control action over the force/torque error [94].

Parallel composition. When using unit quaternion notation, the parallel composition of the desired and the compliant trajectory is defined as

$$\mathbf{p}_{M,r} = \mathbf{p}_{M,c} + \mathbf{p}_{M,d} \quad \dot{\mathbf{p}}_{M,r} = \dot{\mathbf{p}}_{M,c} + \dot{\mathbf{p}}_{M,d} \quad \ddot{\mathbf{p}}_{M,r} = \ddot{\mathbf{p}}_{M,c} + \ddot{\mathbf{p}}_{M,d} \quad (5.38)$$

$$\mathbf{o}_{M,r} = \mathbf{o}_{M,c} * \mathbf{o}_{M,dc} \quad \boldsymbol{\omega}_{M,r}^c = \boldsymbol{\omega}_{M,c}^c + \boldsymbol{\omega}_{M,dc}^c \quad \dot{\boldsymbol{\omega}}_{M,r}^c = \dot{\boldsymbol{\omega}}_{M,c}^c + \dot{\boldsymbol{\omega}}_{M,dc}^c. \quad (5.39)$$

For conflicting situations, the integral control action for the compliant pose ensures dominance of the force control, see (5.37).

5.4 High-Speed Vision

The structure of the vision module is depicted in Fig. 5.14, showing two modules: object tracking and trajectory prediction. In the following, tracking and trajectory prediction of a spherical object is described, as this is part of the experimental scenario in Chapter 6.

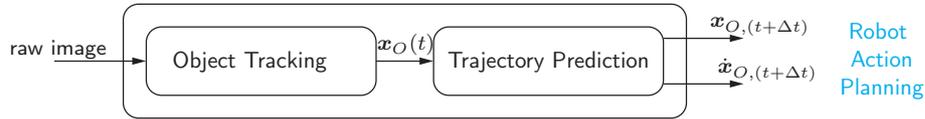


Fig. 5.14: Block diagram of the vision module.

5.4.1 Object Tracking Module

The tracking algorithm assumes that color and shape information of the tracked object are available. The main part of the algorithm is based on the color and shape information of the object. First, a Gaussian smoothing filter is applied to the image in RGB-format. Then, the image is converted into the HSV color space. The HSV space consists of the three channels hue (H), saturation (S), and value (V) and the object is extracted based on the information in the H- and V-channel. For the hue channel, a look-up table is used containing the color information of the tracked object. For the value channel, a threshold value is set to exclude darker regions of the image from the analysis. This is helpful since the darker regions do not provide reliable color information. Combining the two images with an *and* operation results in a binary image. In the next step, an *opening* operation is performed on the binary image in order to filter out speckles, not related edges, and noise in the image [49]. The *opening* includes an elementary erosion operation followed by a dilation operation, both using a 3×3 pixel circle as structural element. Finally, by computing the first order moment of the image, the geometric center of the object is obtained. A block diagram with the processing steps of the tracking algorithm is depicted in Fig. 5.15. In order to increase the tracking frequency and to reduce the latency, a moving window search method is used: once the object is detected in the full size image (1280×1024 pixel), the search area is reduced to a 180×180 pixel window. The position of the window is determined by a linear position prediction and tracking is performed with approximately 150 Hz. If the object is lost, e.g. due to occlusion, the search algorithm is again applied to the full-size image. Further details on the algorithm can be found in [159].

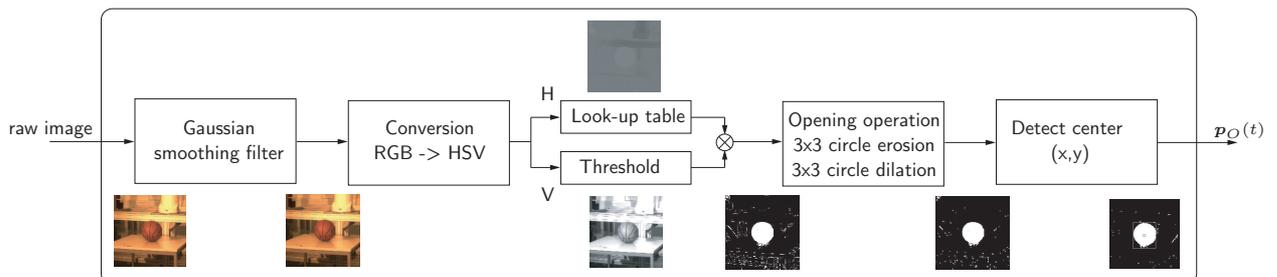


Fig. 5.15: Object tracking module: block diagram of the image processing

5.4.2 Trajectory Prediction Module

The input for the module is the object position $\mathbf{p}_O(t)$ tracked by the camera system. The following two paragraphs describe the trajectory prediction of the object's trajectory for a free-flight phase and for an impact event. Again, a spherical object is considered.

Free-flight. First, the module has to determine whether the object is in free-flight. This can be realized in various ways: one approach is to check whether the vertical object acceleration matches the gravitational acceleration for a certain period of time. If that is the case, one can assume that the object is in free-flight. The drawback of this method is that the acceleration $\mathbf{a}_O(t)$ of the object has to be determined. However, if only position information is available, noisy and hence imprecise acceleration estimates impede this approach. Hence, a different approach is used in this thesis: a free-flight phase is detected when the objects position is outside the workspace of the two robots and the human (which has to be pre-specified).

The prediction of the objects trajectory is realized with a recursive least squares fitting method for a sample period ΔT_s . The trajectory for free-flight is given by

$$\mathbf{p}_{[k]} = \begin{bmatrix} p_{x[k]} \\ p_{y[k]} \\ p_{z[k]} - 0.5gt_{[k]}^2 \end{bmatrix} = \begin{bmatrix} p_{x,0} & v_{x,0} \\ p_{y,0} & v_{y,0} \\ p_{z,0} & v_{z,0} \end{bmatrix} \begin{bmatrix} 1 \\ t_{[k]} \end{bmatrix} \quad (5.40)$$

where g denotes the gravitational acceleration. This results in the recursive least squares estimates for the parameters

$$\hat{\mathbf{m}}_{i[k+1]} = \hat{\mathbf{m}}_{i[k]} + \mathbf{h}_{[k]} (\bar{p}_{i[k+1]} - \mathbf{R}_{[k+1]}^T \hat{\mathbf{m}}_{i[k]}) \quad (5.41)$$

where $\bar{p}_{i[k+1]}$ is the measured i -coordinate of the object and

$$\begin{aligned} \hat{\mathbf{m}}_{i[k]} &= [p_{i,0[k]} \quad v_{i,0[k]}]^T \quad i \in \{x, y, z\}, & \mathbf{R}_{[k+1]} &= [1 \quad t_{[k+1]}]^T \\ \mathbf{h}_{[k]} &= \frac{\mathbf{\Pi}_{[k]} \mathbf{R}_{[k+1]}}{1 + \mathbf{R}_{[k+1]}^T \mathbf{\Pi}_{[k]} \mathbf{R}_{[k+1]}}, & \mathbf{\Pi}_{[k+1]} &= \mathbf{\Pi}_{[k]} - \mathbf{h}_{[k]} \mathbf{R}_{[k+1]}^T \mathbf{\Pi}_{[k]}. \end{aligned} \quad (5.42)$$

With the estimates for the initial state of the object, the future object trajectory $\mathbf{x}_O(t+\Delta t)$, $\dot{\mathbf{x}}_O(t+\Delta t)$ is predicted which serves as input for the robot action planning.

Impacts. For some tasks, it is desirable to predict object trajectories that include impact events in addition to free-flight phases, compare e.g. the indirect catching or the dribbling task. The prediction model is based on the following assumptions: first, the angle of impact with respect to the normal direction is smaller than 30° . In this case, no sliding occurs at the contact point, see [35, 31]. Second, rotational velocities of the object are negligible. With these assumptions, the objects tangential velocity after the impact is approximated as

$$\mathbf{v}_{t,O}^+ \approx \frac{m_{Or}^2}{J_O + m_{Or}^2} \mathbf{v}_{t,O}^- \quad (5.43)$$

where J_O denotes the moment of inertia of the ball. Inserting the expression for a spherical shell, $J_O = \frac{2}{3}m_O r_O^2$, in (5.43) leads to a proportionality constant $c_{r,t} = 0.6$ for the horizontal velocities before and after the impact, compare Sec. 2.6. The translational object velocity after the ground impact is hence defined as

$$\mathbf{v}_O^+ = [c_{r,t} \quad c_{r,t} \quad -c_r]^T \mathbf{v}_O^- \quad (5.44)$$

where c_r is the coefficient of restitution for the ground impact.

The models for the free flight phase and the impact event provide the possibility to predict the future object trajectory which is then used for the robot action planning.

5.5 Summary

This chapter discussed environment perception and interaction control for dynamic object manipulation. Clearly, precise motion/interaction control and accurate sensor feedback are essential for the execution of such tasks on real robotic hardware. However, the dynamic nature of the tasks poses particular challenges that need to be considered.

In Section 5.2, a sensor fusion approach for improved interaction control and estimation of environment forces and torques was presented. The continuous- and discrete-time system model was derived using quaternion notation for the description of the tool orientation. Based on the system model, the design of six DOF force/torque observers was discussed. Here, designs based on the EKF and on the UKF were presented. For both designs, two different measurement vectors have been considered: the first one uses pose and F/T measurements while the second one also uses acceleration and (rotational) velocity measurements. The four filter designs were first evaluated in a six DOF simulation scenario. Particular interest was paid to the robustness of the filter designs towards changing environment conditions. To this end, the simulation was performed in three different conditions: a perfectly known, an overestimated, and an underestimated measurement noise. Six error measures were introduced to evaluate the performance for force and torque estimation during continuous and discrete phases. Despite the decline in performance, the variation of the measurement noise showed that the filters can significantly improve the contact force/torque estimation even for inaccurate estimates of the measurement noise levels. For the observer designs with extended measurement vector, the underestimation of the measurement noise was less critical than the overestimation. For the designs with pose and F/T measurements, the effects of over- and underestimation were comparable. In addition to the simulation, a six DOF experiment was conducted to verify the results. For both, simulation and experiment, the UKF with the extended measurement vector performed best. Although it showed in most of cases only small improvements compared to the EKF with extended measurements, it offers the additional benefit that the calculation of process and measurement jacobians is not needed which makes the UKF easier to implement.

In Section 5.3, the motion and interaction control schemes for the robotic manipulators were discussed. The motion control is realized in task space with the well-established inverse dynamics approach. For the interaction control, a direct force control scheme based

on parallel composition is used. To meet the demands of dynamic object manipulation, the scheme was augmented with the dynamic force/torque observer which provides the estimated environment forces and torques for the interaction control.

In Section 5.4, a method for object tracking with high frame rates (appr. 150 Hz) was detailed. In addition to the tracking algorithm, the section discussed the trajectory prediction for spherical objects. The prediction is based on the models discussed in Chapter 2 and was presented for free-flight phases and impact events as these are of particular interest for the case studies considered in this thesis. However, the prediction can be used in a similar way for other motion phases, such as rolling or sliding.

6 Experimental Evaluation

Summary Experimental results of dynamic object manipulation are presented: nonprehensile catching, throwing, dribbling, and juggling tasks are performed with a six DOF industrial robot and a 14 DOF anthropomorphic dual-arm manipulator. The results of the previous chapters with respect to modeling, environment perception, planning, and control design are integrated in a robotic basketball demonstration scenario. In addition, the benefits of a compliant end effector design are demonstrated by comparing the performance of the compliant and the rigid end effector design for the dribbling task. Finally, the experiments provide valuable insights into practical issues that have not been considered in the theoretical analysis.

The experimental evaluation focuses on robotic basketball as demonstration scenario. Basketball is an excellent challenge to study dynamic and interactive manipulation: the game incorporates intrinsically dynamic manipulation tasks such as catching, throwing, and dribbling. The execution of these tasks requires a high degree of mobility, a detailed environment perception, and a precise timing of motions. Furthermore, the scenario demonstrates exemplarily that dexterous manipulation skills can be realized with simplified end effector designs. Equipping robots with such a *dynamic dexterity* is particularly helpful for the handling of objects that are oversized for conventional grippers.

The chapter is organized as follows: Section 6.1 introduces the hardware setup used for the experiments. Then, Section 6.2 presents results of the throwing experiments that have been performed with the dual-arm manipulator. Next, Section 6.3 details the nonprehensile catching experiments that have been conducted. Both, direct and indirect catching have been realized in experiments. Section 6.4 shows experimental results for ball dribbling and considers two different end effector designs: dribbling with a rigid end effector is detailed in Subsection 6.4.1 and dribbling with a compliant end effector is presented in Subsection 6.4.2. A comparison of the two dribbling tasks is presented in Subsection 6.4.3. Finally, Section 6.5 shows results for the classic juggling tasks performed with the dual-arm manipulator in open-loop control.



Fig. 6.1: Experimental setup: prototype of a compliant end effector design (left), Stäubli six DOF industrial robot (middle), and the dual-arm robot with 14 DOF (right).

6.1 Experimental Setup

For the experiments, a six DOF industrial robot and a 14 DOF dual-arm robotic manipulator are used, see Fig. 6.1. The robots are equipped with either six DOF force/torque sensors or twelve DOF force/torque and acceleration sensors from JR3 that are located at the wrist. Circular plates are used as end effectors. The plates are attached to robot either through a rigid connection or through an elastic coupling which allows relative motion between plate and robot. The developed end effector design with intrinsic compliance is shown in Fig. 6.1.

The stereo vision system for object tracking consists of two Mikrotрон MC1311 high-speed cameras, two frame grabbers and a general purpose PC with two PCIe ports. The tracking algorithm obtains images from the frame grabber for image processing. The object position is sent to the control PC via a TCP network connection. The two cameras are mounted with a baseline $b = 2$ m and converging axes, the distance between baseline and fixation point is 3 m.

The overall control structure was presented in Chapter 5, see Fig. 5.1 for a schematic.

6.2 Throwing

In order to evaluate the throwing strategy, a basketball was placed stationary in the workspace of an anthropomorphic dual-arm robot with 14 DOF. The robot then grasped the ball with a force closure grasp. The trajectory planning for the task was detailed in Subsec. 3.3.2. Experimental snapshots of ball throwing with the dual-arm manipulator are depicted in Fig. 6.2. With the current manipulator hardware, the throwing distance is limited: for target points which are further away than 3 m from the robot base, all generated trajectories violate dynamic constraints (e.g. maximum end effector acceleration). However, for target points within this range the trajectory generation finds feasible solutions. During task execution, the dynamic force sensing is used to apply a constant grasping force and to maintain the contact between the ball and the two end effectors.

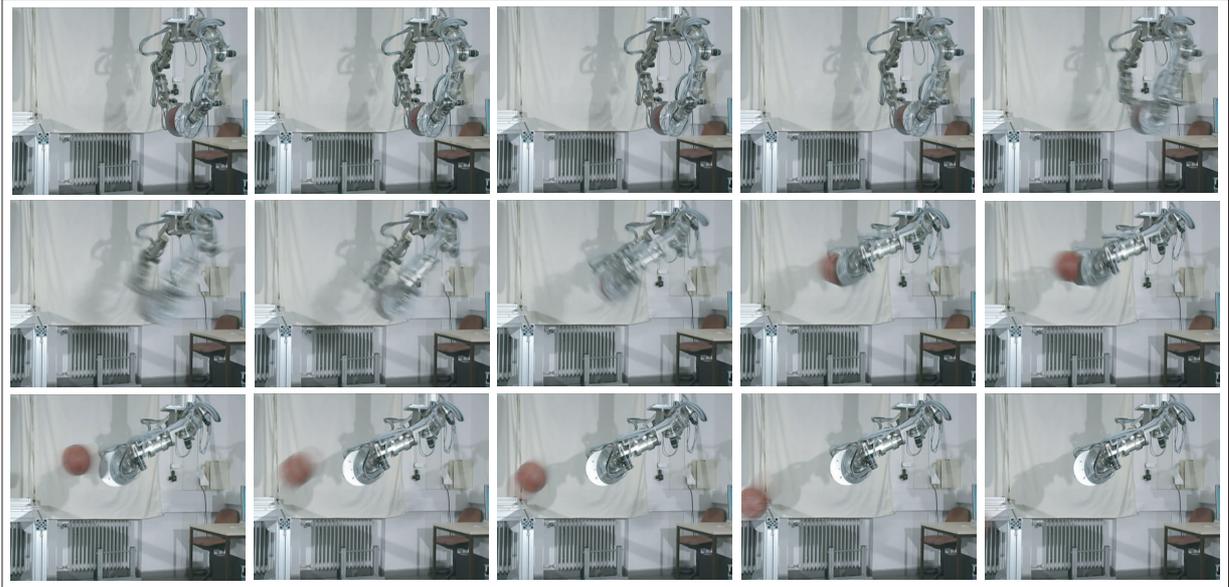


Fig. 6.2: Throwing - experimental snapshots: motion sequence of the dual-arm manipulator from 0 s to 1 s.

6.3 Nonprehensile Catching

The catching task was performed with the six DOF industrial robot. It is initiated when the human operator throws a basketball into the workspace of the robot. In the following, experimental results for the ball trajectory prediction and the two nonprehensile catching methods are presented.

Trajectory prediction. Once the ball is detected by the camera system, it is tracked with appr. 150 Hz and the trajectory is predicted based on the recursive least squares algorithm presented in Subsec. 5.4.2. If a free-flight phase is detected for 100 ms, the prediction is used to create the end effector trajectory for direct or indirect catching which took additional 20 ms. The results of the ball trajectory prediction for the vertical z - and horizontal y -direction for two different sample periods ΔT_s are illustrated in Fig. 6.3. For both, y - and z -direction, the accuracy is significantly improved when using the longer sample period: for a prediction horizon of $\Delta t = 0.4$ s, the deviations from the tracked position are typically less than 0.1 m ($\Delta T_s = 50$ ms) respectively 0.02 m ($\Delta T_s = 100$ ms).

Direct catch. A snapshot sequence of a direct catch is shown in Fig. 6.4. For $t < t_c$, the robot adjusts its position and velocity to match the constraints at t_c (snapshots 3-6), compare Subsec. 3.3.3. Then, for $t > t_c$, the ball is decelerated and the balancing scheme controls the end effector orientation based on the force/torque measurements (snaps. 7-15).

Indirect catch. The ball and effector trajectories for an indirect catch are illustrated in Fig. 6.5. For the initial contact, the end effector adjusts its orientation according to (3.18)

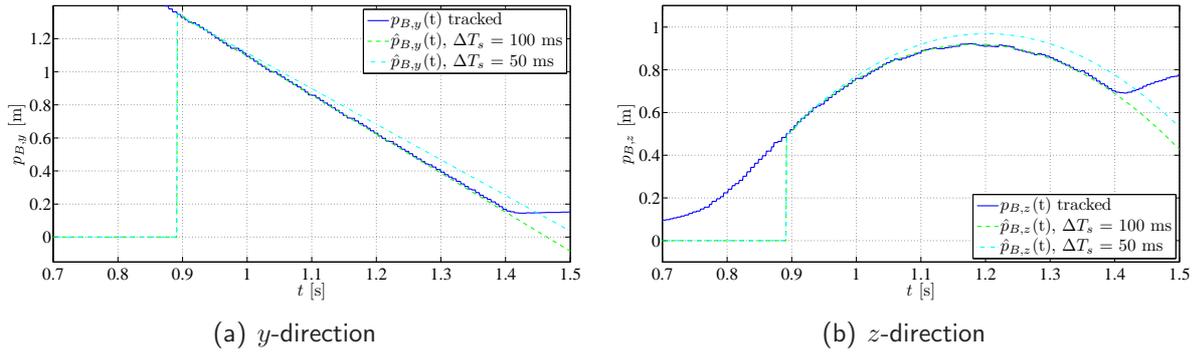


Fig. 6.3: Nonprehensile catching - actual and predicted ball trajectory for different sample periods ΔT_S .

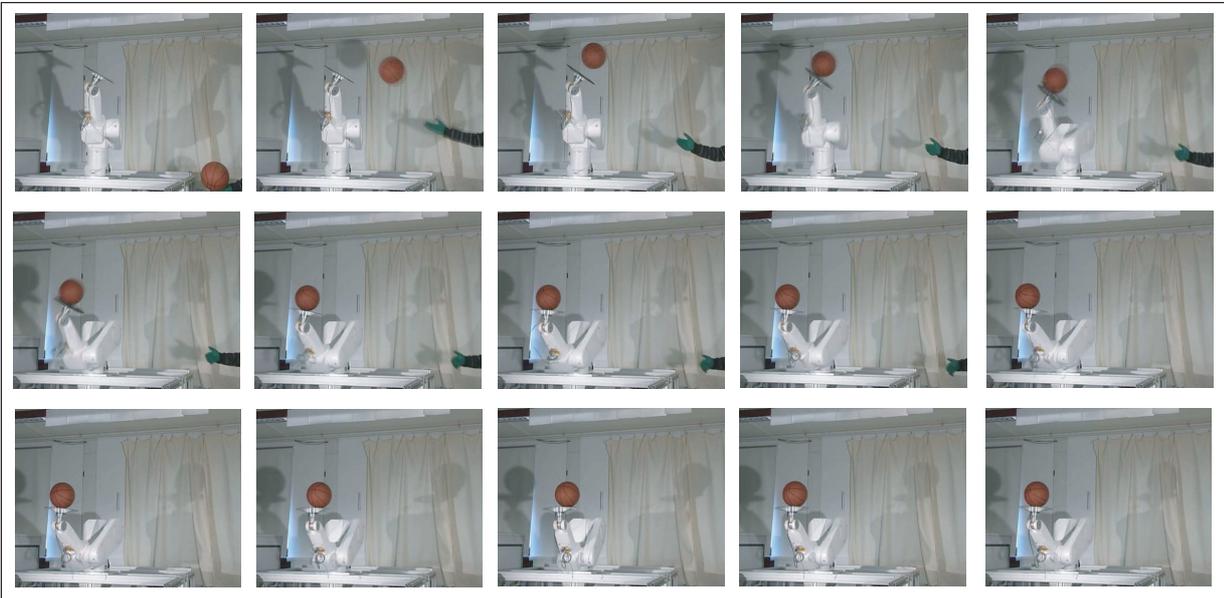


Fig. 6.4: Direct catch - experimental snapshots: initialization by human operator, catching and balancing (sequence from 0 s to 3 s).

to let the ball rebound vertically. The ball is then caught with the second contact at t_c . Fig. 6.5(d) shows a sequence of snapshots for an indirect catch: After the free flight phase (snaps. 2-4), the initial contact occurs at t_r (snaps. 5), the ball rebounds vertically (snaps. 6-7) and a continuous contact is established at t_c (snaps. 8). Then, the ball is decelerated and balanced on the plate. Further details on the implementation and the experiments are given in [156, 158].

Discussion. With the current hardware setup, a direct catch is only dynamically feasible for a small range of ball trajectories. However, if a ball trajectory is within this range, the success rate of the algorithm is more than 80%. A catch is considered successful if the ball comes to rest on the plate. While an indirect catch can be realized for a larger range of ball trajectories, its success rate is only 20%. Here, the underlying modeling assumptions

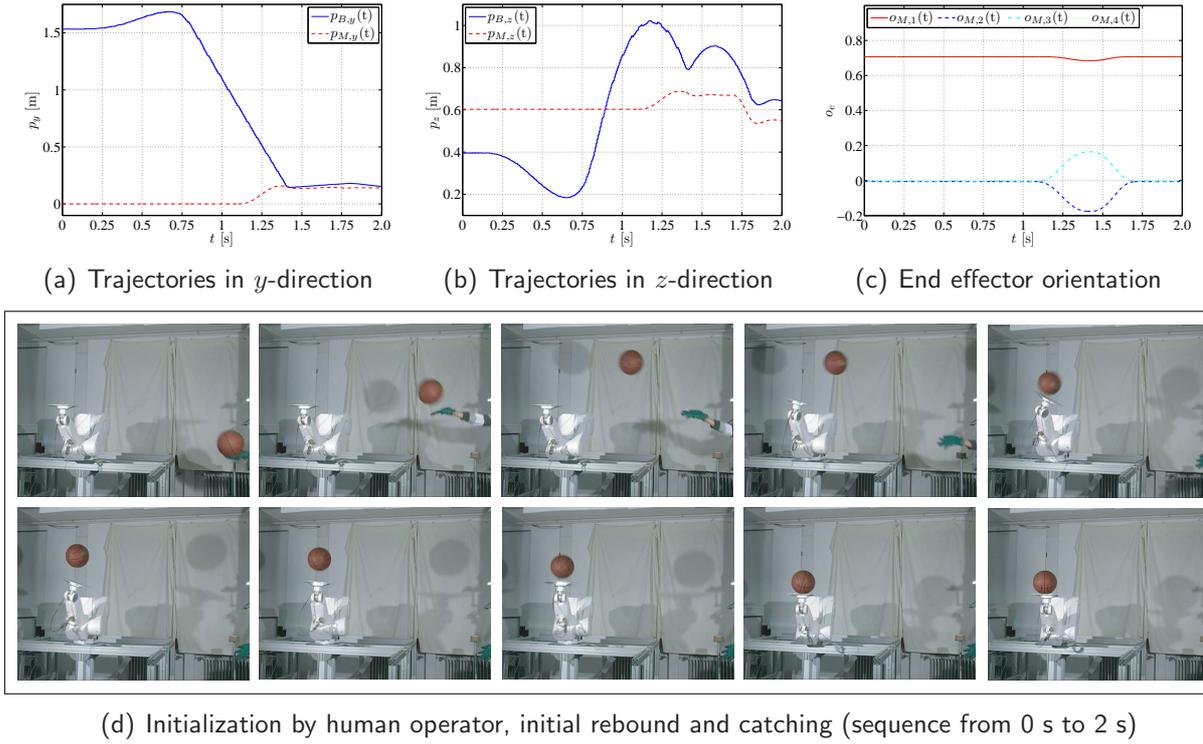


Fig. 6.5: Indirect catch - ball and end effector trajectories: (a) horizontal y -direction, (b) vertical z -direction, (c) end effector orientation, and (d) experimental snapshots.

of the trajectory generation do not describe the impact event accurately, compare Sec. 3.3. Clearly, the range of feasible ball trajectories is influenced by various factors and can be improved by modifying the field of view of the camera system or the manipulator hardware.

6.4 Dribbling

For the analysis of the dribbling task in Subsec. 4.5.2 and Sec. 4.6, simplifying assumptions were made: ball and manipulator motion were restricted to the vertical direction, impacts were modeled as instantaneous inelastic collisions described by c_r , and rotational ball velocity was assumed to be negligible. For the hardware experiments, these modeling errors must be considered and their effects have to be compensated.

The inputs for the robot trajectory generation are the predicted ball position in the xy -plane $p_{B,x}(t_i)$ and $p_{B,y}(t_i)$, the predicted ball velocity $\mathbf{v}_B(t_i^-)$ at the impact / dribbling height h_d , and the point in time t_i when h_d is reached.

Trajectory prediction. The creation of a desired end effector trajectory is based on a prediction of the ball trajectory. As depicted in Fig. 6.6(a) and 6.6(b), the two different methods described in Subsec. 5.4.2 are evaluated: a prediction before and a prediction after the bounce on the ground. In Fig. 6.6(a), the value of the variable changes from 0 to h_d when the next prediction of t_i becomes available. Accordingly, a step down from h_d to 0 occurs once the predicted time t_i is reached. This means, the wider the step, the earlier

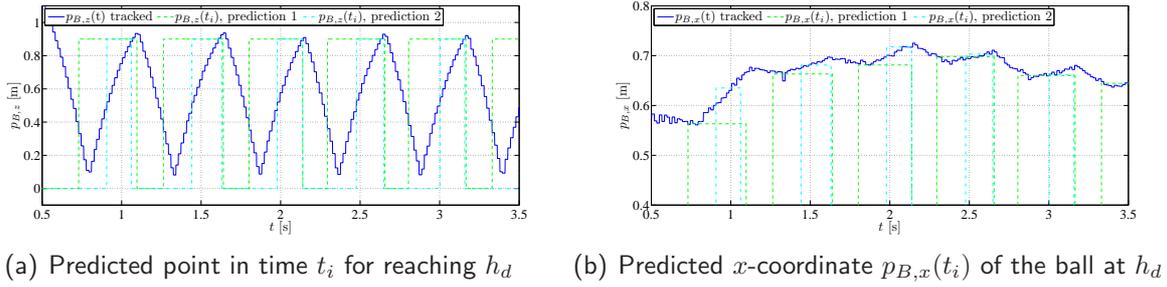


Fig. 6.6: Dribbling with a rigid end effector - actual and predicted ball trajectory for two prediction horizons.

the prediction is available and for an ideal prediction, the step down will occur exactly when the ball reaches h_d . As depicted in Fig. 6.6(a), the accuracy of both approaches is almost identical while the first approach provides the predicted value approximately 0.15 s earlier. Fig. 6.6(b) shows that, for the prediction of the x -coordinate of the ball at the dribbling height, the second approach is more accurate. Again, the drawback of the second method is that it requires a faster motion execution of the robot since the time span until the ball reaches the dribbling height h_d is approximately 0.15 s smaller. In the presented experiments, the robot trajectory is generated based on the first prediction method.

Initialization. The dribbling task can be initialized either by the human operator or the robot. In the former case, the task is triggered when the ball is dropped into a specified area of the robot workspace. Such an initialization is shown in Fig. 6.8. In the latter case, the ball is initially at rest on the plate and then dropped by the robot to start the dribbling cycle. This *autonomous* initialization is illustrated in Fig. 6.9.

6.4.1 Rigid End Effector

Compensation of horizontal deviations. Horizontal deviations from the desired position are compensated by adjusting the orientation of the end effector. The orientation is described by quaternion representation $\{\cos(\phi/2), \mathbf{r}_a \sin(\phi/2)\}$, with axis \mathbf{r}_a in the horizontal xy -plane and angle $\phi = 0$ corresponding to an end effector plate facing the ground.

To determine the required horizontal ball velocity after the actuated impact $v_{B,h}(t_i^+)$, the deviations e_x and e_y of the ball's x - and y -coordinate and the overall error e_h in the horizontal plane at the specified dribbling height

$$e_x = p_{B,x,des}(t_i) - p_{B,x}(t_i) \quad e_y = p_{B,y,des}(t_i) - p_{B,y}(t_i) \quad e_h = \sqrt{e_x^2 + e_y^2} \quad (6.1)$$

are calculated, see Fig. 6.7. Based on the deviations e_x and e_y , the axis of rotation

$$\mathbf{r}_a = \frac{1}{e_h} [-e_y \quad -e_x \quad 0]^T \quad (6.2)$$

is defined. The horizontal motion of the ball during one cycle is the sum of the contributions

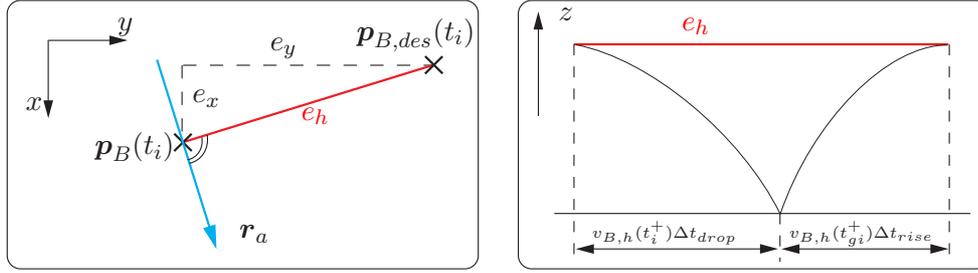


Fig. 6.7: Dribbling with a rigid end effector - position error e_h and axis of rotation \mathbf{r}_a .

in drop and rise phase

$$e_h = v_{B,h}(t_i^+) (\Delta t_{drop} + c_{r,t} \Delta t_{rise}), \quad (6.3)$$

which defines the required horizontal ball velocity $v_{B,h}(t_i^+)$. With the assumption of no sliding between the contact points during impact and neglecting rotational velocity, the vertical and horizontal ball velocity after the impact are determined as a function of ϕ and the known inputs. With small angle approximation, one obtains

$$\begin{aligned} v_{B,z}(t_i^+) &\approx -c_r v_{B,z}(t_i^-) + (1 + c_r) v_{M,z}(t_i) + \phi (c_{r,t} - c_r) v_{B,h}(t_i^-) \\ v_{B,h}(t_i^+) &\approx \phi (c_{r,t} + c_r) (v_{B,z}(t_i^-) - v_{M,z}(t_i)) + c_{r,t} v_{B,h}(t_i^-). \end{aligned} \quad (6.4)$$

With (6.1)-(6.4), the vertical end effector velocity $v_{M,z}$ and the rotation angle ϕ are determined. At time t_i for which the ball is predicted to reach the dribbling height h_d , the desired position, orientation, and velocity of the end effector are

$$\begin{aligned} \mathbf{p}_{M,des}(t_i) &= [p_{B,x}(t_i) \quad p_{B,y}(t_i) \quad h_d + r_B]^T, \\ \mathbf{o}_{M,des}(t_i) &= \{\cos(\phi/2), \mathbf{r}_a \sin(\phi/2)\} * \mathbf{o}_{e,i}, \\ \mathbf{v}_{M,des}(t_i) &= [0 \quad 0 \quad v_{M,z}(t_i)]^T. \end{aligned} \quad (6.5)$$

After the impact with the ball, the end effector decelerates and returns to its initial/rest height. During the dribbling cycle, the horizontal end effector position is adjusted to match the tracked horizontal ball position. For further details on the implementation and a comparison of the performance between vision-based and force/torque-based ball tracking, see Bätz et al. [151].

Dribbling cycle. The ball is held in a human's hand and then dropped into a specified area of the robot workspace to initialize the dribbling. With vision-based ball tracking, dribbling for multiple cycles (> 15) is achieved. Fig. 6.8 shows a sequence of snapshots taken during an experiment. The time duration between two consecutive snapshots is 0.16 s. The ball and end effector trajectory in vertical direction are depicted in Fig. 6.11(a).

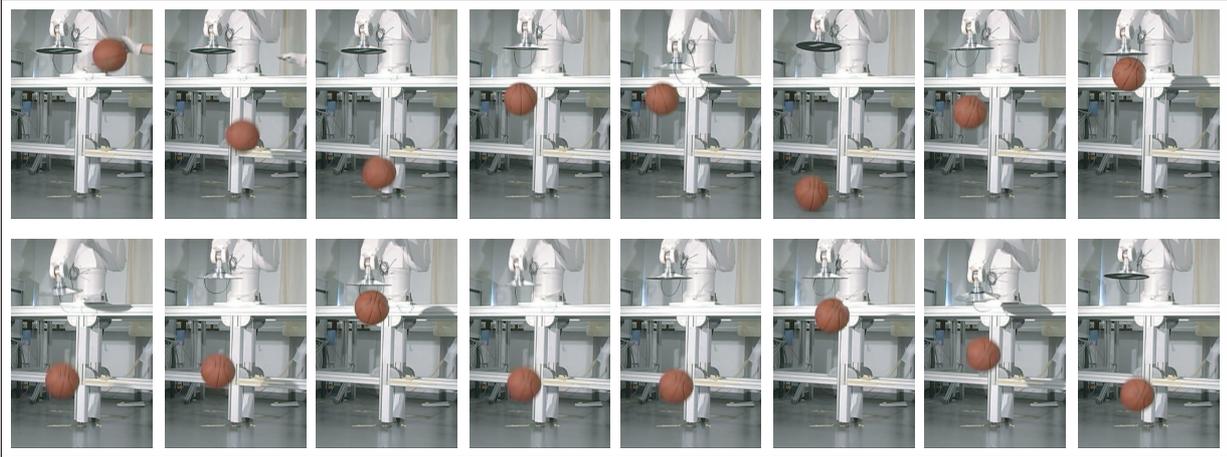


Fig. 6.8: Dribbling with a rigid end effector - experimental snapshots: initialization by a human operator and four dribbling cycles (sequence from 0.5 s to 2.9 s).

6.4.2 Compliant End Effector

The end effector design with intrinsic compliance is depicted in Fig. 6.1. The springs allow to store energy temporarily and release it at the end of contact, leading to possible higher accelerations. The design reduces the effective inertia to 0.3 kg during initial contact/impact, allowing for a continuous contact phase.

Dribbling cycle. The desired initial and final states for the experimentally studied dribbling cycle are summarized in Table 6.1. For the experiment, a constraint for the relative velocity between ball and manipulator at the time of initial contact is added:

$$\dot{x}_M(t_f^c) = 0.95\dot{x}_B(t_f^c). \quad (6.6)$$

For the contact phases *catch* and *push*, the control law described in Sec. 4.6 is used. During the non-contact phase the robot trajectory is generated and updated based on the tracked ball position and the predicted impact time. For all phases, an additional controller is used to compensate horizontal deviations. Fig. 6.11(b) shows the actual manipulator and ball position in vertical direction during a dribbling experiment. The ball trajectory starts at 0.75 s when the ball is detected by the vision system. A snapshot sequence of the initialization and the first dribbling cycle is depicted in Fig. 6.9.

The performance is mainly limited by the following aspect: in the simulation, it was assumed that there is no energy loss in the elastic actuator. For the real actuator however, friction forces during spring compression/elongation cause energy dissipation. This, in turn, leads to a reduced amount of energy storage in the spring and thus a reduced energy transfer to the ball.

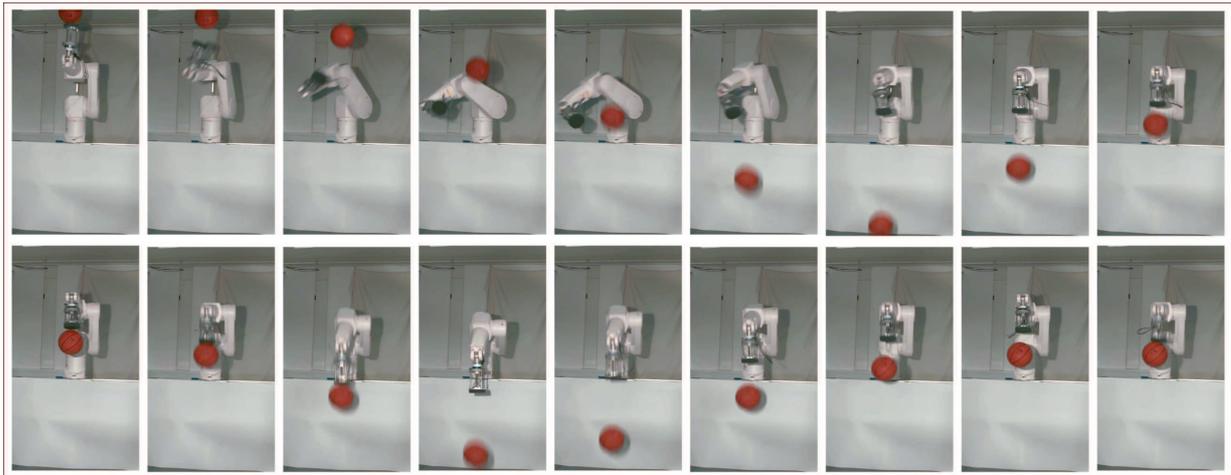


Fig. 6.9: Dribbling with a compliant actuator - experimental snapshots: autonomous task initialization and first dribbling cycle (sequence from 0 s to 2.16 s).

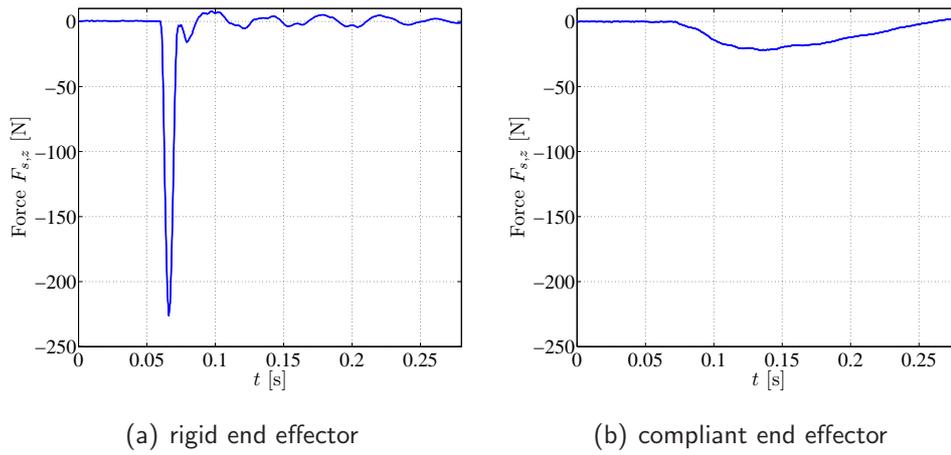
	initial contact	switch	release
x_M [m]	1.605	1.65*	1.285
x_B [m]	1.32	1.38	1.00
\dot{x}_M [m/s]	1.1438	0	-2.91*
\dot{x}_B [m/s]	1.204	0	-4.26

Tab. 6.1: Initial and final states for the experimentally studied dribbling cycle. Entries marked with * are obtained through trajectory optimization.

6.4.3 Comparison

Reduced mechanical load. The use of a mechanical spring element promises to reduce the mechanical load on the end effector. In order to evaluate this assumption, the following experiment was conducted: a basketball impacted the end effector with a relative velocity of 1.4 m/s. Fig. 6.10(a) shows the contact force for a rigid end effector design: during the initial impact, the force exceeds 220 N. In contrast, Fig. 6.10(b) shows the contact force for the compliant end effector design: here, the maximum force is significantly reduced (appr. 25 N). In addition, the compliant structure leads to a continuous contact phase after the initial impact.

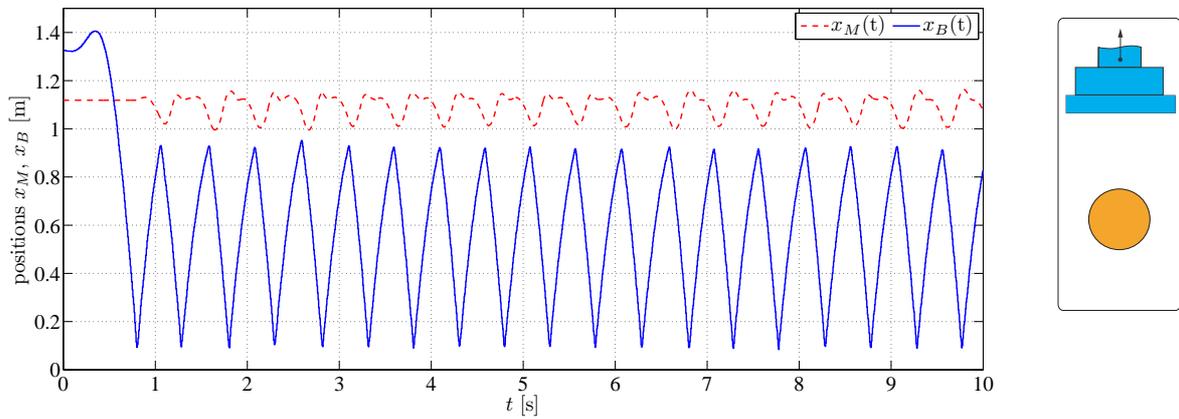
Contact phase. Fig. 6.11 depicts the vertical position of manipulator and ball during a dribbling sequence of ten seconds for a (a) rigid and (b) compliant end effector. The ball trajectories show significant differences: with the rigid end effector, the contact phase reduces to an impact event. The time duration of approximately 20 ms justifies the modeling assumption of an *instantaneous* impact, compare Sec. 4.5. This is illustrated by the discontinuous change in the ball velocity when contact with the manipulator occurs. In contrast, with the compliant end effector, discontinuities of the ball velocity only occur for the ground impacts. The instantaneous impact between end effector and ball is now replaced with a continuous contact with a catch and push phase as discussed in Subsec. 4.6.



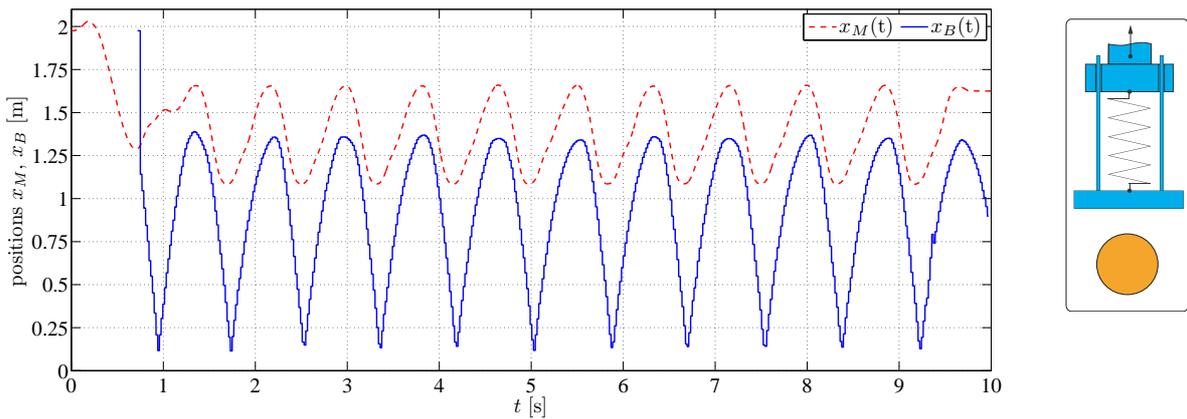
(a) rigid end effector

(b) compliant end effector

Fig. 6.10: Measured impact forces for (a) rigid end effector and (b) compliant end effector.



(a) rigid end effector



(b) compliant end effector

Fig. 6.11: Actual manipulator and ball position in vertical direction for a dribbling sequence of ten seconds.

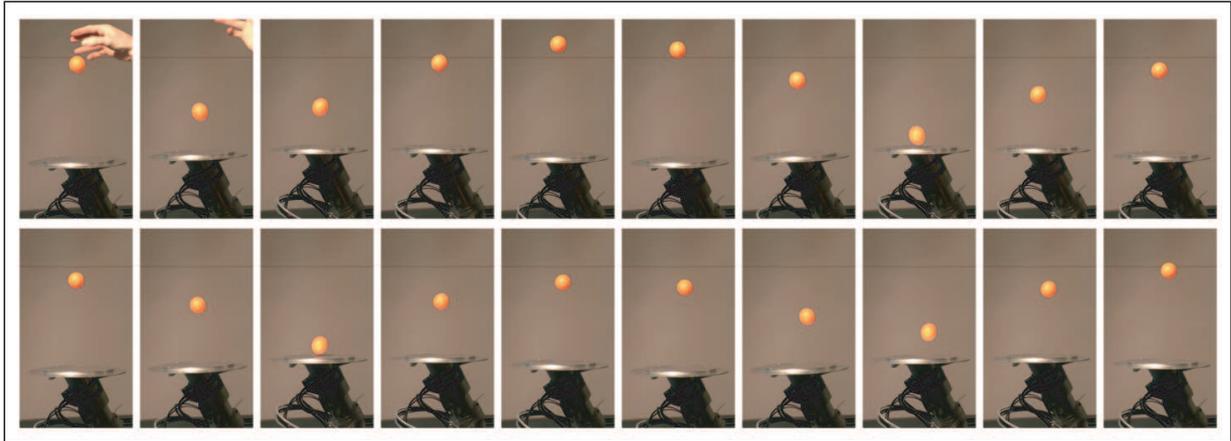


Fig. 6.12: Juggling - experimental snapshots: motion sequence of ball and dual-arm manipulator.

6.5 Juggling

The juggling of a ping-pong ball was experimentally studied using the dual-arm manipulator. The task was performed in open-loop control and the manipulator trajectory was optimized based on the approach presented in Subsec. 4.5.1.

Fig. 6.12 shows a snapshot sequence of the juggling task: the initial state of the systems is determined by the human operator. If \boldsymbol{x}_0 is within the region of attraction, the system converges to the desired fixed point \boldsymbol{x}^* . The coefficient of restitution of the ping-pong ball is appr. $c_r = 0.9$. Such a value of c_r close to 1 reduces the rate of convergence to the desired periodic orbit, compare Subsec. 4.5.1. However, the region of attraction is large enough so that nine of ten task initializations lead to a stable periodic motion.

Horizontal deviations from the fixed point are compensated by using a slightly curved end effector plate with curvature $K = 1 \text{ m}^{-1}$. Further details on this approach are given in [102]. A similar effect can be realized either with a compliant structure comparable to a membrane or by adjusting the orientation of the end effector based on sensor feedback, see Subsec. 6.4.1.

Fig. 6.13(a) shows the measured forces in vertical direction $F_{S,z}$. Since inertial forces have a non-negligible effect on these measurements, it is not possible to determine the forces that are exchanged with the environment. Clearly, this effect becomes even more pronounced if the juggling height and/or the ball mass were reduced. In contrast, Fig. 6.13(b) shows the estimated environment forces $F_{E,z}$ using the UKF-based force/torque observer with \boldsymbol{y}_2 , compare Sec. 5.2. Here, the inertial effects have been compensated and the impact forces can be directly identified.

6.6 Summary

This chapter presented the experimental results for a number of tasks that require robots with a high degree of dynamic dexterity. The successful execution of the tasks validates

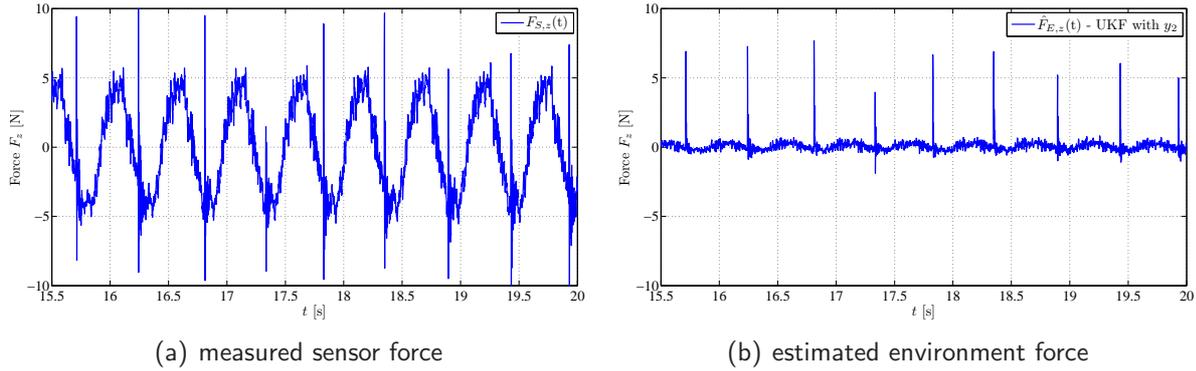


Fig. 6.13: Juggling: (a) measured sensor forces $F_{S,z}$ and (b) estimated environment force $F_{E,z}$ (UKF with y_2).

the results of the previous chapter: the offline optimization of trajectories and the creation of look-up tables is an excellent approach to obtain both, near-optimal trajectories, and real-time capability with fast reaction times. The control framework presented in Ch. 5 allows a thorough environment perception. The source and/or amount of sensor feedback can be chosen task-dependent: the catching task was performed using both, visual and F/T-sensor information. For the dribbling task, only visual feedback was used whereas the throwing motions were executed based on F/T-sensor information. Finally, the juggling was performed in open-loop control based on the trajectory optimization discussed in Subsec. 4.5.1.

Clearly, the mechanical design of a robotic manipulator is of great importance for its dynamic dexterity. While the hardware design is not the main focus of this thesis, the comparison of the dribbling task for two end effectors provides valuable implications for future hardware design. While using the same manipulator, the addition of an elastic element has a number of beneficial effects: first, the kinetic energy of the ball can be transformed in potential energy of the spring. This, in turn, facilitates the task execution as it relaxes the acceleration requirements for the manipulator. Second, the elastic element also reduces the mechanical load for the manipulator by establishing a continuous-time contact phase that replaces the instantaneous impact with impulsive contact forces. Third, the continuous-time contact phase facilitates the compensation of horizontal deviations (in the xy -plane) from the desired fixed point.

The experiments in this chapter were limited to the manipulation of spherical objects. This simplified the image processing as the tracking of the objects orientation is dispensable. However, the extension of the presented approach for other object geometries is straightforward: the trajectory prediction for the orientation can be realized in the same way as for the position. The offline optimization and the creation of look-up tables require only small modifications: as discussed in Subsec. 3.2, the orientation of the object has to be used as additional optimization criteria.

7 Conclusions

The chapter provides a summary and discussion of the material presented in this work. In addition, possible directions of future research work are outlined.

7.1 Summary

The following paragraphs briefly summarize the contributions of the previous chapters.

Modeling foundations for dynamic object manipulation. Chapter 2 provided a summary and a discussion of state-of-the-art models that lay the foundations for dynamic object manipulation. A hybrid system model was introduced to capture the task dynamics with varying contact states. The contact kinematics were outlined in a general form and it was shown how the equations simplify for special cases such as rolling or sliding motions. In addition, the chapter discussed the fundamental properties of static and dynamic friction models. Finally, the modeling of impact events was considered and both, discrete and continuous, impact models were outlined.

Planning methods for dynamic manipulation tasks. The realization of dynamic object manipulation with a generic end effector design was considered.

First, Chapter 3 discussed non-periodic dynamic manipulation tasks. It was detailed how dynamic dexterity can be realized with generic end effector designs and the planning of optimal trajectories was discussed. The proposed approach was based on a combination of offline and online decisions: selection criteria were evaluated offline and used for the creation of look-up tables. These look-up tables then allowed online evaluation of the selection criteria. Three non-periodic manipulation tasks were exemplarily studied: rolling manipulation, dual-handed throwing, and one-handed catching. For robotic catching, a novel nonprehensile approach was discussed which allows the catching of bulky objects. Two catching strategies, direct and indirect, were proposed to increase the range of admissible initial object states.

Next, Chapter 4 focused on periodic manipulation tasks with intermittent contact. Ball dribbling was introduced as novel case study for dynamic dexterity and the hybrid system dynamics, optimal trajectory planning, and control design were presented. A comparison with the classic juggling task showed that the dribbling task is more challenging due to the occurrence of an additional, autonomous impact. In addition, the classic juggling task was studied and a new approach for optimal trajectory planning based on a non-local stability analysis was presented.

Dynamic contact force/torque observer. In Chapter 5, a sensor fusion approach for the estimation of environment forces and torques was developed. The design of six

DOF force/torque observers based on the EKF and on the UKF was discussed. For both approaches, two measurement vectors were considered: the first one used pose and force/torque measurements while the second one also used acceleration and velocity measurements. The four filter designs were evaluated in a six DOF simulation scenario and in experiments. The obtained results illustrated the importance of such an observer, not only for dynamic object manipulation but also for other tasks that require environment interaction during dynamic motions. With the quantitative comparisons, the appropriate observer design can be chosen according to the task-specific requirements.

Control framework for dynamic object manipulation. Chapter 5 also addressed the challenges for the closed-loop control of dynamic manipulation tasks and presented an extensive control framework to overcome these challenges. With respect to environment perception, a method for high-speed image processing was presented. This allows to track and to predict the state of manipulated objects during non-contact phases with high sampling rates. Together with the dynamic force/torque observer, the high-speed image processing was integrated into the control architecture and used for the robot action planning. In addition, the motion and interaction control was discussed. The force/torque observer was integrated in a direct force control scheme to improve the interaction control during dynamic motions.

Dynamic manipulation with a compliant end effector. The thesis considered the concept of series elastic actuation to improve the performance in dynamic manipulation tasks. While such an approach adds complexity to the task planning, it also promises several beneficial effects: first, kinetic energy of the object can be transformed in potential energy of the spring element. This relaxes the acceleration requirements for the manipulator and facilitates the task execution. Second, the elastic element reduces the mechanical load for the manipulator at impact events as it establishes a continuous-time contact phase which replaces the impulsive contact force of an instantaneous impact. Finally, the continuous-time contact phase extends the time in which the object can be directly controlled with the manipulator. The approach was evaluated in experiments: the dribbling task was performed with both, a compliant and a rigid end effector design. Their comparison confirmed the expected benefits and provides a basis for further development of the mechanical design.

Experimental evaluation. In Chapter 6, nonprehensile catching, throwing, dribbling, and juggling tasks were experimentally studied with a six DOF industrial robot and a 14 DOF anthropomorphic dual-arm manipulator. The experiments validated the approach of the thesis by integrating the results with respect to modeling, environment perception, planning and control design in a robotic basketball scenario. In addition, the experiments provided valuable insights into practical issues that had not been considered in the theoretical analysis.

7.2 Discussion and Future Directions

Individual aspects of the topics presented in this thesis were discussed at the end of each chapter. This section comments on the most important issues and outlines possible improvements and extensions for future research.

Mechanical design. The occurrence of motion sequences with high manipulator acceleration is a characteristic feature of dynamic object manipulation. Consequently, an increase of the peak power is desirable. One way to realize this is the use of motors with higher output power. However, this is not reasonable since size and weight of a humanoid or autonomous robot are usually constrained and, in addition, a *temporary* increase is sufficient. Hence, the concept of series elastic actuation was employed in this work. The benefits were exemplarily demonstrated by comparing the dribbling task with a rigid and a compliant end effector design. On the one hand, the results emphasized the importance of the mechanical design for the planning and execution of dynamic manipulation tasks. On the other hand, the experiments showed that the *optimal* stiffness is task-dependent and can even change for different phases of a particular task. Thus, a manipulator design with variable stiffness actuation (VSA) is desirable to further improve the system performance. Such an adjustable stiffness is also beneficial for manipulation tasks that include a physical human-robot interaction.

Manipulation of elastic and arbitrarily shaped objects. The manipulated objects in this work were effectively rigid, compare Sec. 2.6. This allowed the use of discrete impact models and hence simplified the task planning and trajectory optimization. For the planning of dynamic manipulation tasks with elastic objects, continuous or distributed models are needed. Furthermore, the applications and case studies in this work considered the manipulation of spherical objects. Thus, the object orientation did not influence the task planning. The manipulation of non-spherical objects generally implies preferred orientations and hence poses additional challenges for both task planning and object tracking.

Multi-camera system. In this work, a stereo camera system was utilized to provide visual feedback of the environment. With this approach, an occlusion of the object in one camera results in a loss of 3D information. In order to address this problem, future research work could focus on a multi-camera setup that allows robust tracking of the object at high frame rates. Another important criteria for successful task execution is the field of view of the cameras, compare Sec. 6.3. With a multi-camera system, different field of views can be fused to obtain a thorough visual perception of the environment.

Library of dynamic manipulation skills. The planning and control methods presented in this work were applied to a number of case studies. However, the modeling foundations, the planning approach, and the overall control structure can be used for a wide range of tasks which require a dynamically dexterous robot. Hence, by building upon this framework, future research work can establish a library of dynamic manipulation skills.

Dynamic manipulation with multi-fingered robotic hands. This work considered the realization of dynamic manipulation tasks with simple end effector designs. Such an approach is desirable as it generalizes to more complex hands, e.g. when only the palm of a multi-fingered hand is used for a particular task. However, dynamic dexterity and multi-fingered hands clearly do not exclude each other. On the contrary, some dynamic manipulation tasks even require hands with multiple DOF, e.g. twisting a pen in one's hand. Realizing dynamic object manipulation with a multi-fingered hand is one of the future milestones for robotic manipulation.

A Appendix

A.1 Differential Geometry

Detailed information on the following terms can be found in the in the books by DoCarmo and Murray [34, 92].

First fundamental form. For a surface, the first fundamental form describes how the inner product of two tangent vectors is related to the natural inner product on \mathbb{R}^3 . For a local parameterization $\mathbf{c}(u, v)$, the matrix representation of the first fundamental form at a point \mathbf{p} is given by

$$\mathbf{I}_p = \begin{bmatrix} \mathbf{c}_u^T \mathbf{c}_u & \mathbf{c}_u^T \mathbf{c}_v \\ \mathbf{c}_v^T \mathbf{c}_u & \mathbf{c}_v^T \mathbf{c}_v \end{bmatrix}. \quad (\text{A.1})$$

Metric tensor. The metric tensor of a surface is given by the square root of the first fundamental form

$$\mathbf{I}_p = \mathbf{M}_p \mathbf{M}_p. \quad (\text{A.2})$$

For an orthogonal parameterization, \mathbf{M}_p takes the form

$$\mathbf{M}_p = \begin{bmatrix} \|\mathbf{c}_u\| & 0 \\ 0 & \|\mathbf{c}_v\| \end{bmatrix}. \quad (\text{A.3})$$

Outward pointing unit normal vector. The vector

$$\hat{\mathbf{n}}(u, v) = \frac{\mathbf{c}_u \times \mathbf{c}_v}{\|\mathbf{c}_u \times \mathbf{c}_v\|} \quad (\text{A.4})$$

denotes the unit normal at a point of the surface.

Second fundamental form. For a surface, the second fundamental form is defined as

$$\mathbf{II}_p = \begin{bmatrix} \mathbf{c}_u^T \hat{\mathbf{n}}_u & \mathbf{c}_u^T \hat{\mathbf{n}}_v \\ \mathbf{c}_v^T \hat{\mathbf{n}}_u & \mathbf{c}_v^T \hat{\mathbf{n}}_v \end{bmatrix}. \quad (\text{A.5})$$

Curvature tensor. For an orthogonal parameterization, the curvature tensor is defined as

$$\mathbf{K}_p = \mathbf{M}_p^{-T} \mathbf{II}_p \mathbf{M}_p^{-1} = \begin{bmatrix} \frac{\mathbf{c}_u^T \hat{\mathbf{n}}_u}{\|\mathbf{c}_u\|^2} & \frac{\mathbf{c}_u^T \hat{\mathbf{n}}_v}{\|\mathbf{c}_u\| \|\mathbf{c}_v\|} \\ \frac{\mathbf{c}_v^T \hat{\mathbf{n}}_u}{\|\mathbf{c}_u\| \|\mathbf{c}_v\|} & \frac{\mathbf{c}_v^T \hat{\mathbf{n}}_v}{\|\mathbf{c}_v\|^2} \end{bmatrix}. \quad (\text{A.6})$$

Torsion form. The torsion of a surface is a measure of how the Gauss frame twists when moving across the surface. It is defined as

$$\mathbf{T}_p = \begin{bmatrix} \frac{\mathbf{c}_v^T \mathbf{c}_{uv}}{\|\mathbf{c}_u\|^2 \|\mathbf{c}_v\|} & \frac{\mathbf{c}_v^T \mathbf{c}_{vv}}{\|\mathbf{c}_u\| \|\mathbf{c}_v\|^2} \end{bmatrix}. \quad (\text{A.7})$$

Geometric parameters of a surface. For a given parameterization, the collective of the metric tensor \mathbf{M} , the curvature \mathbf{K} , and the torsion form \mathbf{T} are the geometric parameters of the surface.

A.2 Velocity and Force/Torque Transformations

Velocity transformation. The velocities of two points A and C fixed on a rigid body are related according to the following relation

$$\dot{\mathbf{r}}_C^0 = \dot{\mathbf{r}}_A^0 + \boldsymbol{\omega}_A^0 \times \mathbf{r}_{AC}^0 \quad (\text{A.8})$$

$$\boldsymbol{\omega}_C^0 = \boldsymbol{\omega}_A^0, \quad (\text{A.9})$$

or in matrix notation

$$\begin{bmatrix} \dot{\mathbf{r}}_C^0 \\ \boldsymbol{\omega}_C^0 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{S}(\mathbf{r}_{AC}^0) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_A^0 \\ \boldsymbol{\omega}_A^0 \end{bmatrix}. \quad (\text{A.10})$$

When the velocities are referred to their own frames one obtains

$$\begin{bmatrix} \dot{\mathbf{r}}_C^C \\ \boldsymbol{\omega}_C^C \end{bmatrix} = \begin{bmatrix} \mathbf{R}_A^C & -\mathbf{R}_A^C \mathbf{S}(\mathbf{r}_{AC}^A) \\ \mathbf{0} & \mathbf{R}_A^C \end{bmatrix} \begin{bmatrix} \dot{\mathbf{r}}_A^A \\ \boldsymbol{\omega}_A^A \end{bmatrix}. \quad (\text{A.11})$$

Here, the subscripts A and C for $\boldsymbol{\omega}$ can be omitted since the rotational velocity is identical for every point on the rigid body.

Force/Torque transformation. The forces and torques applied on a rigid body at point A can be transformed to corresponding forces and torques at a different reference point C

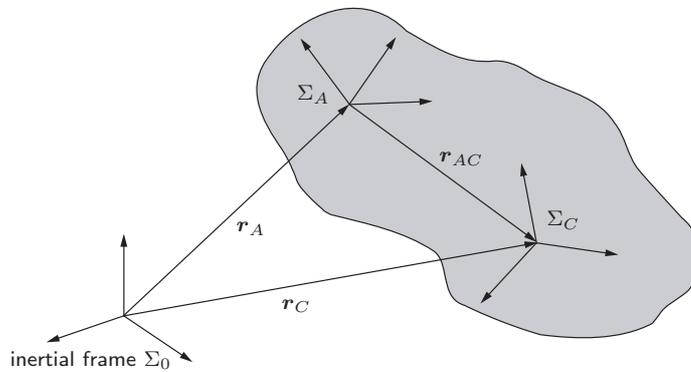


Fig. A.1: Representation of velocities in different coordinate frames.

according to

$$\begin{bmatrix} \mathbf{F}_C^0 \\ \mathbf{M}_C^0 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{S}(\mathbf{r}_{CA}^0) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_A^0 \\ \mathbf{M}_A^0 \end{bmatrix}. \quad (\text{A.12})$$

When the forces and torques are referred to their own frames one obtains

$$\begin{bmatrix} \mathbf{F}_C^C \\ \mathbf{M}_C^C \end{bmatrix} = \begin{bmatrix} \mathbf{R}_A^C & \mathbf{0} \\ \mathbf{S}(\mathbf{r}_{CA}^C)\mathbf{R}_A^C & \mathbf{R}_A^C \end{bmatrix} \begin{bmatrix} \mathbf{F}_A^A \\ \mathbf{M}_A^A \end{bmatrix}. \quad (\text{A.13})$$

In some cases, however, it will be helpful to illustrate the *origin* of the transformed forces and torques. Then, the following extended notation is used

$$\begin{bmatrix} \mathbf{F}_{A \rightarrow B}^C \\ \mathbf{M}_{A \rightarrow B}^C \end{bmatrix}, \quad (\text{A.14})$$

which means the forces and torques applied at point A have been transformed to the corresponding forces and torques at point B and are expressed in frame C .

A.3 Orientation of a Rigid Body: Unit Quaternion

This section outlines the use of unit quaternions to describe the orientation of a rigid body. A detailed discussion on unit quaternions and on other generalized coordinates for the orientation of a rigid body can be found in the books by Wittenburg and Natale [145, 94].

Skew-symmetric operator. The skew-symmetric operator is defined as

$$\mathbf{S}(\mathbf{r}) = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \quad (\text{A.15})$$

Unit quaternion. The unit quaternion is defined as

$$\mathbf{o} = \{\eta, \boldsymbol{\epsilon}\}, \quad (\text{A.16})$$

where η is the scalar part and $\boldsymbol{\epsilon} = [\epsilon_x \ \epsilon_y \ \epsilon_z]^T$ is the vector part of the quaternion. The unit quaternion fulfills the condition

$$\|\mathbf{o}\|^2 = \eta^2 + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = \eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = 1. \quad (\text{A.17})$$

The inverse of the unit quaternion is given by

$$\mathbf{o}^{-1} = \{\eta, -\boldsymbol{\epsilon}\}. \quad (\text{A.18})$$

The quaternion product operation is defined as

$$\mathbf{o}_1 * \mathbf{o}_2 = \begin{bmatrix} \eta_1 \eta_2 - \boldsymbol{\epsilon}_1^T \boldsymbol{\epsilon}_2 \\ \eta_1 \boldsymbol{\epsilon}_2 + \eta_2 \boldsymbol{\epsilon}_1 + \mathbf{S}(\boldsymbol{\epsilon}_1) \boldsymbol{\epsilon}_2 \end{bmatrix}. \quad (\text{A.19})$$

Angular velocity. An expression for the angular velocity can be derived based on (A.19). Starting with an initial orientation \mathbf{o}_{10} at time t , the rotation \mathbf{o}_{20} is interpreted as a differential rotation over the time interval dt ,

$$\mathbf{o}_{20} = \{\eta_{20}, \boldsymbol{\epsilon}_{20}\} = \left\{ \cos\left(\frac{1}{2} \|\boldsymbol{\omega}\| dt\right), \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \sin\left(\frac{1}{2} \|\boldsymbol{\omega}\| dt\right) \right\}. \quad (\text{A.20})$$

For $dt \rightarrow 0$, $\mathbf{o}_{20} = \{1, \frac{1}{2}\boldsymbol{\omega}\}$ and (A.19) can be rewritten as

$$\eta(t + dt) = \eta_{10}(t) - \frac{1}{2} \boldsymbol{\epsilon}_{10}(t)^T \boldsymbol{\omega} dt, \quad (\text{A.21})$$

$$\boldsymbol{\epsilon}(t + dt) = \boldsymbol{\epsilon}_{10}(t) + \frac{1}{2} [\eta_{10}(t) \boldsymbol{\omega} + \mathbf{S}(\boldsymbol{\omega}) \boldsymbol{\epsilon}_{10}(t)] dt. \quad (\text{A.22})$$

This yields to the so called unit quaternion propagation

$$\dot{\eta}_{10} = -\frac{1}{2} \boldsymbol{\epsilon}_{10}^T \boldsymbol{\omega}_{10}, \quad (\text{A.23})$$

$$\dot{\boldsymbol{\epsilon}}_{10} = \frac{1}{2} [\eta_{10} \mathbf{I} - \mathbf{S}(\boldsymbol{\epsilon}_{10})] \boldsymbol{\omega}_{10}. \quad (\text{A.24})$$

A.4 Newton-Euler Equations for a Rigid Body

For a reference point A fixed on the rigid body and an inertial frame Σ_0 as reference frame, the Newton-Euler equations for a rigid body are given by [145]

$$\begin{aligned} \mathbf{F}_A &= m \ddot{\mathbf{r}}_A - m \mathbf{S}(\mathbf{r}_{AC}) \boldsymbol{\omega} + m \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{AC}) \\ \mathbf{M}_A &= m \mathbf{r}_{AC} \times \ddot{\mathbf{r}}_A + \mathbf{J}_A \dot{\boldsymbol{\omega}}_A + \boldsymbol{\omega}_A \times \mathbf{J}_A \boldsymbol{\omega}_A \end{aligned} \quad (\text{A.25})$$

If the reference point A coincides with the center of mass $\mathbf{r}_{AC} = 0$, (A.25) simplifies to

$$\begin{aligned} \mathbf{F}_A &= m \ddot{\mathbf{r}}_A \\ \mathbf{M}_A &= \mathbf{J}_A \dot{\boldsymbol{\omega}}_A + \boldsymbol{\omega}_A \times \mathbf{J}_A \boldsymbol{\omega}_A \end{aligned} \quad (\text{A.26})$$

Inertia tensor. The inertia tensor of a rigid body with respect to a body-fixed reference point A and expressed in the inertial frame Σ_0 is given by

$$\mathbf{J}_A = \mathbf{J}_A^0. \quad (\text{A.27})$$

If the rigid body is rotating, the inertia tensor expressed in the inertial frame is time-varying. A constant expression for the inertia tensor is obtained by choosing a body-fixed

reference frame, such as a frame Σ_C located at the center of mass (COM),

$$\mathbf{J}_A^C. \tag{A.28}$$

For a constant reference point A , a change of the reference frame for the inertia tensor is realized by

$$\mathbf{J}_A^C = \mathbf{R}_B^C \mathbf{J}_A^B \mathbf{R}_C^B. \tag{A.29}$$

Bibliography

- [1] SMERobot - The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing. <http://www.smerobot.org>, 2011.
- [2] E.W. Aboaf, S.M. Drucker, and C.G. Atkeson. Task-level robot learning: Juggling a tennis ball more accurately. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1290 – 1295, 1989.
- [3] R.L. Andersson. Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6):728–739, 1989.
- [4] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083 –1138, 1994.
- [5] C. N. Bapat, S. Sankar, and N. Popplewell. Repeated impacts on a sinusoidally vibrating table reappraised. *Journal of Sound and Vibration*, 108(1):99–115, 1986.
- [6] B. Bäuml, T. Wimböck, and G. Hirzinger. Kinematically optimal catching a flying ball with a hand-arm-system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2592–2599, 2010.
- [7] L. Biagiotti and C. Melchiorri. *Advances in Telerobotics*, chapter Environment Estimation in Teleoperation Systems, pages 211–232. Springer (Berlin / Heidelberg), 2007.
- [8] A. Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road towards simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000.
- [9] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 348–353, 2000.
- [10] A. Bicchi and V. Kumar. *Lecture Notes in Control and Information Science*, chapter Robot Grasping and Manipulation, pages 55–74. Springer (Berlin / Heidelberg), 2001.
- [11] A. Bicchi and A. Marigo. Rolling contacts and dexterous manipulation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 282–287, 2000.
- [12] A. Bicchi, D. Prattichizzo, and S. Sastry. Planning motions of rolling surfaces. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 2812–2817, 1995.

-
- [13] P.-A. Bliman and M. Sorine. Easy-to-use realistic dry friction models for automatic control. In *Proc. European Control Conference (ECC)*, pages 3788–3794, 1995.
- [14] B. Bona and M. Indri. Friction compensation in robotics: an overview. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 4360–4367, 2005.
- [15] R. M. Brach. Formulation of rigid body impact problems using generalized coefficients. *International Journal of Engineering Science*, 36(1):61–71, 1998.
- [16] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, 1998.
- [17] R. Brockett. *Essays on Control: Perspectives in the Theory and its Applications*, chapter Hybrid models for motion control systems, pages 29–53. Birkhäuser (Basel), 1993.
- [18] B. Brogliato. *Nonsmooth Mechanics - Models, Dynamics, and Control*. Springer (Berlin / Heidelberg), 1999.
- [19] B. Brogliato and A. Zavalo-Rio. On the control of complementary-slackness juggling mechanical systems. *IEEE Transactions on Automatic Control*, 45(2):235–246, 2000.
- [20] M. Bühler, D.E. Koditschek, and P.J. Kindlmann. A one degree of freedom juggler in a two degree of freedom environment. In *Proc. IEEE International Workshop on Intelligent Robots*, pages 91–97, 1988.
- [21] M. Bühler, D.E. Koditschek, and P.J. Kindlmann. A family of robot control strategies for intermittent dynamical environments. *IEEE Control Systems Magazine*, 10(2):16–22, 1990.
- [22] M. Bühler, D.E. Koditschek, and P.J. Kindlmann. Planning and control of robotic juggling and catching tasks. *International Journal of Robotics Research*, 13(2):101–118, 1994.
- [23] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–555, 1999.
- [24] R.R. Burridge, A.A. Rizzi, and D.E. Koditschek. Toward a dynamical pick and place. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 292–297, 1995.
- [25] M. Buss. Methoden zur Regelung Hybrider Dynamischer Systeme - Modelle, Regelkreise, Optimalsteuerung, Rechnerwerkzeuge und Mechatronik-Anwendungen. Habilitationsschrift, 2000.
- [26] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Six-dof impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation*, 15(2):289–300, 1999.

- [27] C. Canudas de Wit, H. Olsson, K. J. Aström, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.
- [28] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.
- [29] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. 4(3):273–285, 1999.
- [30] E. Coumans. The bullet physics simulation toolkit. www.bulletphysics.com.
- [31] R. Cross. Grip-slip behavior of a bouncing ball. *American Journal of Physics*, 11:1093–1102, 2002.
- [32] P. R. Dahl. Solid friction damping of mechanical vibrations. In *Proc. AIAA Guidance and Control Conference*, 1976.
- [33] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.
- [34] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall (New Jersey), 1976.
- [35] A. Domenech. A classical experiment revisited: The bounce of balls and superballs in three dimensions. *American Journal of Physics*, 1:28–36, 2005.
- [36] M. A. Erdmann. exploration of nonprehensile two-palm manipulation: Planning and execution. In *International Symposium on Robotics Research*, 1995.
- [37] H. Frank, D. Barteit, N. Wellerdick-Wojtasik, T. Frank, G. Novak, and S. Mahlknecht. Autonomous mechanical controlled grippers for capturing flying objects. In *Proc. IEEE International Conference on Industrial Informatics*, pages 431–436, 2007.
- [38] U. Frese, B. Bäuml, S. Haidacher, G. Schreiber, I. Schäfer, M. Hähnle, and G. Hirzinger. Off-the-shelf vision for a robotic ball catcher. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1623–1629, 2001.
- [39] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa. Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 181–187, 2006.
- [40] M. Gäfvert. Comparisons of two dynamic friction models. In *Proc. IEEE International Conference on Control Applications*, 1997.
- [41] J. A. Gangloff and M. F. de Mathelin. High-speed visual servoing of a 6-d.o.f. manipulator using multivariable control. *Advanced Robotics*, 17(10):993–1021, 2003.

-
- [42] J. Garcia, A. Robertsson, J. Ortega, and R. Johansson. Generalized contact force estimator for a robot manipulator. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 4019–4024, 2006.
- [43] J.G. Garcia, A. Robertsson, J.G. Ortega, and R. Johansson. Sensor fusion for compliant robot motion control. *IEEE Transactions on Robotics*, 24(2):430–441, 2008.
- [44] K. J. Gergen and M. M. Gergen. *Social Psychology*. Springer (Berlin / Heidelberg), 1986.
- [45] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Mechanism and Machine Theory*, 37(10):1213–1239, 2002.
- [46] C. Glocker. *Dynamik von Starrkörpersystemen mit Reibung und Stößen*. PhD thesis, Technische Universität München, 1995.
- [47] A. Haddadi and K. Hashtrudi-Zaad. A new method for online parameter estimation of hunt-crossley environment dynamic models. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 981–986, 2008.
- [48] D. Haessig and B. Friedland. On the modeling and simulation of friction. In *Proc. American Control Conference (ACC)*, pages 1256–1261, 1990.
- [49] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 2002.
- [50] H. Hertz. Über die Berührung fester elastischer Körper. *Journal für die reine und angewandte Mathematik*, 92:156–171, 1881.
- [51] I. Hiskens. Stability of hybrid system limit cycles: Application to the compass gait biped robot. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 774–779, 2001.
- [52] J. Hodgins and M. Raibert. Biped gymnastics. *International Journal of Robotics Research*, 9:115–128, 1990.
- [53] N. Hogan. Impedance control: An approach to manipulation. In *Proc. American Control Conference (ACC)*, pages 304–313, 1984.
- [54] P. J. Holmes. The dynamics of repeated impacts with a sinusoidally vibrating table. *Journal of Sound and Vibration*, 84(2):173–189, 1982.
- [55] B. Hove and J.-J. Slotine. Experiments in robotic catching. In *Proc. American Control Conference (ACC)*, pages 380–386, 1991.
- [56] C. S. Hsu. Global analysis by cell mapping. *International Journal of Bifurcation and Chaos*, 2(4):727–771, 1992.
- [57] W. Huang. *Impulsive Manipulation*. PhD thesis, Carnegie Mellon University, 1997.

- [58] K. H. Hunt and F. R. E. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *ASME Journal of Applied Mechanics*, 42(2):440–445, 1975.
- [59] Y. Imai, A. Namiki, K. Hashimoto, and M. Ishikawa. Dynamic active catching using a high-speed multifingered hand and a high-speed vision system. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1849–1854, 2004.
- [60] E. A. Jackson. *Perspectives of Nonlinear Dynamics*, volume 1. Cambridge University Press (Cambridge), 1989.
- [61] H.-L. Jin, Q. Ye, and M. Zacksenhouse. Return maps, parameterization, and cycle-wise planning of yo-yo playing. *IEEE Transactions on Robotics*, 25(2):438–445, 2009.
- [62] H.-L. Jin and M. Zacksenhouse. Oscillatory neural networks for robotic yo-yo control. *IEEE Transactions on Neural Networks*, 14(2):317–325, 2003.
- [63] H.-L. Jin and M. Zacksenhouse. Robotic yoyo playing with visual feedback. *IEEE Transactions on Robotics*, 20(4):736–744, 2004.
- [64] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. Technical report, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997.
- [65] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings Of the IEEE*, 92(3):401–422, 2004.
- [66] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. American Control Conference (ACC)*, pages 1628–1632, 1995.
- [67] N. Kato, K. Matsuda, and T. Nakamura. Adaptive control for a throwing motion of a 2 dof robot. In *Proc. International Workshop on Advanced Motion Control*, pages 203–207, 1996.
- [68] S. Katsumata, S. Ichinose, T. Shoji, S. Nakaura, and M. Sampei. Throwing motion control based on output zeroing utilizing 2-link underactuated arm. In *Proc. American Control Conference (ACC)*, pages 3057–3064, 2009.
- [69] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall (New Jersey), 2000.
- [70] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
- [71] D. Koditschek. *Robot Control: Dynamics, Motion Planning and Analysis*, chapter Dynamically dexterous robots. IEEE Press, 1993.
- [72] I. Kolmanovsky and N. H. McClamroch. Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 15(6):20–36, 1995.

-
- [73] E. Kraft. A quaternion-based unscented kalman filter for orientation tracking. In *Proc. IEEE International Conference of Information Fusion*, pages 47–54, 2003.
- [74] T. Kröger, D. Kubus, and F.M. Wahl. 6d force and acceleration sensor fusion for compliant manipulation control. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2626–2631, 2006.
- [75] D. Kubus, T. Kröger, and F.M. Wahl. Improving force control performance by computational elimination of non-contact forces/torques. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 2617–2622, 2008.
- [76] H. Lankarani and P. Nikravesh. Continuous contact force models for impact analysis in multibody systems. *Nonlinear Dynamics*, 5(2):193–207, 1994.
- [77] H. E. Lehtitiet and B. N. Miller. Numerical study of a billiard in a gravitational field. *Physica*, 21(1):93–104, 1986.
- [78] S.-T. Lin. Force sensing using kalman filtering techniques for robot compliant motion control. *Journal of Intelligent and Robotic Systems*, 18(1):1–16, 1997.
- [79] F. Lombai and G. Szederkenyi. Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator. In *Proc. IEEE International Conference on Mechatronics*, pages 1–6, 2009.
- [80] A. Luo and R. Han. The dynamics of a bouncing ball with a sinusoidally vibrating table revisited. *Nonlinear Dynamics*, 10(1):1–18, 1996.
- [81] K. M. Lynch and C. K. Black. Recurrence, controllability and stability of juggling. *IEEE Transaction on Robotics and Automation*, 17(2):113–124, 2001.
- [82] K. M. Lynch and M. T. Mason. Dynamic underactuated nonprehensile manipulation. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 889–896, 1996.
- [83] K. M. Lynch and M. T. Mason. Dynamic manipulation with a one joint robot. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 359–366, 1997.
- [84] K. M. Lynch and M. T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *International Journal of Robotics Research*, 18(1):64–92, 1999.
- [85] K. M. Lynch, N. Shiroma, H. Arai, and K. Tanie. The roles of shape and motion in dynamic manipulation: the butterfly example. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1958–1963, 1998.
- [86] D. W. Marhefka and D. E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems. *IEEE Transactions on Systems, Man and Cybernetics*, 29(6):566–572, 1999.

- [87] A. Marigo and A. Bicchi. Rolling bodies with regular surface: Controllability theory and applications. *IEEE Transactions on Automatic Control*, 45(9):1586–1599, 2000.
- [88] M. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.
- [89] M. T. Mason. Progress in nonprehensile manipulation. *International Journal of Robotics Research*, 18(11):1129–1141, 1999.
- [90] M.T. Mason and K.M. Lynch. Dynamic manipulation. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 152–159, 1993.
- [91] D. J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7:17–32, 1989.
- [92] R. M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press (Boca Raton), 1994.
- [93] A. Namiki, K. Hashimoto, and M. Ishikawa. A hierarchical control architecture for high-speed visual servoing. *International Journal of Robotics Research*, 22:873–888, 2004.
- [94] C. Natale. *Interaction Control of Robot Manipulators - Six-Degrees-of-Freedom Tasks*. Springer (Berlin / Heidelberg), 2003.
- [95] A. M. Okamura, N. Smaby, and M. R. Cutkosky. An overview of dexterous manipulation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 255 – 262, 2000.
- [96] H. Olsson and K. J. Aström. Observer-based friction compensation. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 4345–4350, 1996.
- [97] H. Olsson, K. J. Aström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky. Friction models and friction compensation. *European Journal of Control*, 4(3):176–195, 1998.
- [98] G. Oriolo, M. Vendittelli, A. Marigo, and A. Bicchi. From nominal to robust planning: The plate-ball manipulation system. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 3175–3180, 2003.
- [99] T. S. Parker and L. O. Chua. *Practical Numerical Algorithms for Chaotic Systems*. Springer (Berlin / Heidelberg), 1989.
- [100] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts*. Wiley-VCH (Weinheim), 2004.
- [101] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.
- [102] P. Reist and R. D’Andrea. Bouncing an unconstrained ball in three dimensions with a blind juggling robot. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1774–1781, 2009.

-
- [103] M. Riley and C.G. Atkeson. Robot catching: Towards engaging human-humanoid interaction. *Autonomous Robots*, 12(1):119–128, 2002.
- [104] A. A. Rizzi and D. E. Koditschek. Progress in spatial robot juggling. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 775–780, 1992.
- [105] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: the spatial two-juggle. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 919–924, 1993.
- [106] A. A. Rizzi and D. E. Koditschek. Toward the control of attention in a dynamically dexterous robot. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 123–130, 1993.
- [107] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: solvable mirror laws. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 2935–2940, 1994.
- [108] R. Ronsse, P. Lefevre, and R. Sepulchre. Open-loop stabilization of 2d impact juggling. In *Proc. IFAC Symposium on Nonlinear Control Systems*, pages 1157–1162, 2004.
- [109] R. Ronsse, P. Lefevre, and R. Sepulchre. Timing feedback control of a rhythmic system. In *Proc. IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC)*, pages 6146–6151, 2005.
- [110] R. Ronsse, P. Lefevre, and R. Sepulchre. Sensorless stabilization of bounce juggling. *IEEE Transactions on Robotics*, 22(1):147–159, 2006.
- [111] R. Ronsse, P. Lefevre, and R. Sepulchre. Rhythmic feedback control of a blind planar juggler. *IEEE Transactions on Robotics*, 23(4):790–802, 2007.
- [112] R. Ronsse and R. Sepulchre. Feedback control of impact dynamics: the bouncing ball revisited. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 4807–4812, 2006.
- [113] T. Sakaguchi, Y. Masutani, and F. Miyazaki. A study on juggling task. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1991.
- [114] S. Schaal and C. G. Atkeson. Open loop stable control strategies for robot juggling. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 913–918, 1993.
- [115] S. Schaal and C. G. Atkeson. Memory-based robot learning. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2928–2933, 8–13 May 1994.
- [116] S. Schaal and C. G. Atkeson. Robot juggling: implementation of memory-based learning. *IEEE Control Systems Magazine*, 14(1):57–71, 1994.

- [117] S. Schaal, D. Sternad, and C. G. Atkeson. One-handed juggling: A dynamical approach to a rhythmic movement task. *Journal of Motor Behavior*, 28(2):165–183, 1996.
- [118] A. L. Schwab and M. Wisse. Basin of attraction of the simplest walking model. In *Proc. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1–9, 2001.
- [119] T. Senoo, A. Namiki, and M. Ishikawa. Ball control in high-speed batting motion using hybrid trajectory generator. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1762–1767, 2006.
- [120] T. Senoo, A. Namiki, and M. Ishikawa. High-speed throwing motion based on kinetic chain approach. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3206–3211, 2008.
- [121] R. Sepulchre and M. Gerard. Stabilization of periodic orbits in a wedge billiard. In *Proc. IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1568–1573, 2003.
- [122] D. Shiokata, A. Namiki, and M. Ishikawa. Robot dribbling using a high-speed multifingered hand and a high-speed vision system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2097–2102, 2005.
- [123] A. Shiriaev, L. Freidovich, and S. Gusev. Transverse linearization for controlled mechanical systems with several passive degrees of freedom. *IEEE Transactions on Automatic Control*, 55:1–13, 2010.
- [124] A. Shiriaev, L. Freidovich, and I. Manchester. Periodic motion planning and analytical computation of transverse linearizations for hybrid mechanical systems. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 4326–4331, 2008.
- [125] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer (Berlin / Heidelberg), 2008.
- [126] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics - Modelling, Planning and Control*. Springer (Berlin / Heidelberg), 2008.
- [127] S. N. Simic, K. Johansson, J. Lygeros, and S. Sastry. Towards a geometric theory of hybrid systems. *Dynamics of Continuous, Discrete and Impulsive Systems, Series B*, 12(5-6):649–687, 2005.
- [128] S. N. Simic, K. H. Johansson, J. Lygeros, and S. Sastry. Structural stability of hybrid systems. In *Proc. European Control Conference (ECC)*, pages 3858–3863, 2001.
- [129] J.-J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall (New Jersey), 1991.
- [130] M. Sobotka. *Hybrid Dynamical System Methods for Legged Robot Locomotion with Variable Ground Contact*. PhD thesis, Technische Universität München, 2007.

-
- [131] M. W. Spong. Partial feedback linearization of underactuated mechanical systems. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 314–321, 1994.
- [132] B. Stanczyk. *Development and Control of an Anthropomorphic Telerobotic System*. PhD thesis, Technische Universität München, 2006.
- [133] D. Sternad, M. Duarte, H. Katsumata, and S. Schaal. Dynamics of a bouncing ball in human performance. *Phys. Rev. E*, 63(1):011902, 2000.
- [134] D. Sternad, M. Duarte, H. Katsumata, and S. Schaal. Bouncing a ball: Tuning into dynamic stability. *Journal of Experimental Psychology: Human Perception and Performance*, 27(5):1163–1184, 2001.
- [135] R. Stribeck. Die wesentlichen Eigenschaften der Gleit- und Rollenlager. *Zeitschrift des Vereins Deutscher Ingenieure*, 46:1341–1348, 1902.
- [136] W. J. Stronge. Rigid body collisions with friction. *Mathematical and Physical Sciences*, 431(1881):169–181, 1990.
- [137] C. Studer. *Augmented time-stepping integration of non-smooth dynamical systems*. PhD thesis, ETH Zürich, 2008.
- [138] J. Tian and Y.-B. Jia. Modeling deformable shell-like objects grasped by a robot hand. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1297–1302, 2009.
- [139] M. Uchiyama and K. Kitagaki. Dynamic force sensing for high-speed robot manipulation using kalman filtering techniques. In *Proc. IEEE International Conference on Decision and Control (CDC)*, pages 2147–2152, 1989.
- [140] M.-W. Ueberle. *Design, Control, and Evaluation of a Family of Kinesthetic Haptic Interfaces*. PhD thesis, Technische Universität München, 2006.
- [141] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153 – 158, 2000.
- [142] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill, 2006.
- [143] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press (Boca Raton), 2007.
- [144] H. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2):161–167, 1966.
- [145] J. Wittenburg. *Dynamics of Multibody Systems*. Springer (Berlin / Heidelberg), 2007.

- [146] H. Wu, T. Zhang, A. Borst, K. Kühnlenz, and M. Buss. An explorative study of visual servo control with insect-inspired reichardt-model. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 345–350, 2009.
- [147] T. Yoshikawa. Analysis and control of robot manipulators with redundancy. In *Proc. International Symposium on Robotics Research*, pages 735–747, 1983.
- [148] T. Yoshikawa. Dynamic manipulability of robot manipulators. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1033–1038, 1985.
- [149] T. Yoshikawa. Dynamic hybrid position/force control of robot manipulators—description of hand constraints and calculation of joint driving force. *IEEE Journal of Ro*, 3(5):386–392, 1987.
- [150] A. Zavalo-Rio and B. Brogliato. Direct adaptive control design for one-degree-of-freedom complementary-slackness jugglers. In *Proc. Automatica*, pages 1117–1123, 2001.

Own Publications and Supervised Student Projects

- [151] G. Bätz, K.-K. Lee, D. Wollherr, and M. Buss. Robot basketball: A comparison of ball dribbling with visual and force/torque feedback. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 514–519, 2009.
- [152] G. Bätz, U. Mettin, A. Schmidts, M. Scheint, D. Wollherr, and A. Shiriaev. Ball dribbling with an underactuated continuous-time control phase: Theory & experiments. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2890–2895, 2010.
- [153] G. Bätz, M. Scheint, and D. Wollherr. Towards dynamic manipulation for humanoid robots: Experiments and design aspects. *International Journal of Humanoid Robotics*, 8(3):513–532, 2011.
- [154] G. Bätz, M. Sobotka, D. Wollherr, and M. Buss. Robot basketball: Ball dribbling - a modified juggling task. In T. Kröger and F. M. Wahl, editors, *Advances in Robotic Research - Theory, Implementation, Application*, pages 323–335. Springer, 2009.
- [155] G. Bätz, D. Wollherr, K. Kühnlenz, and M. Buss. *Advances in Real-Time Systems*, chapter Robot Basketball: A New Challenge for Real-Time Control. Springer (Berlin / Heidelberg), 2011.
- [156] G. Bätz, A. Yaqub, D. Wollherr, and M. Buss. Dynamic manipulation: Nonprehensile ball catching. In *Proc. IEEE Mediterranean Conference on Control and Automation (MED)*, pages 365–370, 2010.
- [157] L. Kniep. Dynamische Manipulation mit dem Dual-Arm Manipulator. Master’s thesis, Technische Universität München, 2009.

-
- [158] Kwang-Kyu Lee, G. Bätz, and D. Wollherr. Basketball robot: Ball-on-plate with pure haptic information. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 2410–2415, 2008.
- [159] X. Lu. Aufbau eines Stereo-Kamerasystems und bildbasiertes Tracken schnell bewegter Objekte. Master’s thesis, Technische Universität München, 2008.
- [160] U. Mettin, A. S. Shiriaev, G. Bätz, and D. Wollherr. Ball dribbling with an under-actuated continuous-time control phase. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 4669–4674, 2010.