Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

# Hybrid System Stabilization and Robot Motion Planning for Robust Catching

**Markus Michael Schill**

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:**
Prof. Dr.-Ing. Ulrich Wagner

**Prüfende der Dissertation:**
1. Prof. Dr.-Ing./Univ. Tokio habil. Martin Buss
2. Prof. Dr.-Ing. Alin Albu-Schäffer

Die Dissertation wurde am 19.11.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 06.06.2019 angenommen.

# Foreword

This thesis summarizes my research conducted at the Chair of Automatic Control Engineering, LSR, at the Technical University of Munich, TUM, Germany. I will always be grateful for this chance to freely evolve in close collaboration with colleagues, who not only love to share their expert knowledge of diverse technical domains, but are also supportive in all life situations.

First of all, I would like to thank my thesis advisor Prof. Martin Buss for the opportunity to work as an associate in his team of excellent researchers. His continuous support and trust in my abilities encouraged me to constantly challenge the state of the art, which has lead to the results presented in this thesis. Furthermore, I want to thank Prof. Ulrich Wagner and Prof. Alin Albu-Schäffer for their feedback on this work. I would also like to thank Dr. Marion Leibold for her frequent guidance during my introductory phase, the superior proofreading of my publication drafts and for passing her teaching experience to associates like myself.

I thank the whole team at LSR for all the great moments we had together. An additional warm thanks goes to my colleagues from the Shrine team, who have become friends as we spent year of our lives together: Sotiris Apostolopoulos, Philine Donner, Stefan Friedrich, Stefan Kersting, Alexander Pekarovskiy and Michaela Semmler.

This thesis was also supported and influenced by the commitment of my students Hannah Bernauer, Urs Borrmann, Christian Czempinski, Michael Dirix, Meiqiong Wang and most notably Felix Gruber, Victor Gaßmann and Tanja Weber.

Finally, I thank my parents Waltraud and Johannes, my sister Bettina and my great-uncle Prof. Jörg Eberspächer for their love, support and endless encouragement throughout my entire time of studies and research at TUM.

Marbach am Neckar, August 2019 Markus Schill

## Abstract

Robots catching an object is a challenging and frequently considered testbed to demonstrate the performance of object tracking combined with motion planning in highly dynamic environments. A successful catch relies on effective solutions in both tracking and planning. However, this thesis focuses on the latter, underdeveloped motion planning problem. To manage the catching problem's complexity, a preparatory contribution in this thesis decouples the problem into two conceptual dimensions: Task motion planning (task level) and robot motion planning (joint level) for a physical distinction. Moreover, both dimensions are divided into four distinct temporal phases.

A novel parametrization is then derived to reduce the three-dimensional (3-D) catching problem to one dimension (1-D) on the ballistic flight path. Inversely, an efficient dynamical system formulation allows for reconstruction of solutions from 1-D to 3-D. Hence, the body of work in hybrid dynamical systems theory, in particular on the 1-D bouncing ball problem, becomes available for robotic catching. Uniform Zeno asymptotic stability from the bouncing ball literature is adapted as an example and extended to enable provable robust catching. A quantitative stability measure and the importance of the initial relative state between an object and end-effector are discussed. As a result, constrained dynamic optimization maximizes convergence speed and enables a quantitative success-oriented comparison of catching motions.

Catching motions are characterized by high velocities and accelerations, which require robot operation close to the velocity, torque and power limitations of motors and gears. Not exceeding these limitations is vital for guaranteeing the validity of the aforementioned success-oriented comparison in real applications. Hence, this thesis proposes an augmented kinematic formulation for nonprehensile manipulation through intermittent contacts occurring in catching, batting or juggling. In such scenarios, the contact point with an end-effector is variable and is proposed to be modeled with additional virtual joints at the end of the kinematic chain. While not in contact with the manipulated part, these new joints are unconstrained with regard to velocity and acceleration. An optimization-based, and thus tuning-free, comparison of differential inverse kinematic approaches is conducted, given that the manipulation task's path or trajectory are known. Simulations reveal that the proposed augmentation enables dynamically feasible acceleration variations at high velocities on and close to a given path.

The feasible and optimal solutions at the task and joint level are ultimately validated for two symmetrical robots autonomously playing throw and catch. This controllable robot-robot scenario enables repeatable experimental results and a statistically significant number of trials per setting.

## Zusammenfassung

Robotern das Fangen von Objekten beizubringen, ist ein herausforderndes und doch regelmäßig gewähltes Anwendungsbeispiel. Es dient gerne zur Funktionalitätsdemonstration in höchst dynamischen Umgebungen für Lösungsansätze, welche Objekterfassung und Bewegungsplanung kombinieren. Vor dem Hintergrund, dass Forschung bisher ihren Fokus überwiegend auf die Objekterfassung legte, widmet sich diese Arbeit vorrangig Fragestellungen der Bewegungsplanung mit einem Fokus auf robuster Aufgabenerfüllung. Um die Komplexität des Fangproblems zu bewältigen, unterteilt diese Arbeit die Problemstellung zunächst anhand zweier konzeptioneller Dimensionen: die Aufgabenbewegungsplanung (*task level*) und die Roboterbewegungsplanung (*joint level*) als physikalische Unterscheidung. Die beiden konzeptionellen Dimensionen werden wiederum je in vier Phasen zur zeitlichen Differenzierung unterteilt.

Auf dieser Basis wird eine neuartige Parametrierung hergeleitet um das dreidimensionale (3-D) Fangproblem auf eine Dimension (1-D) entlang der ballistischen Flugbahn zu reduzieren. Um aus 1-D Lösungen wieder 3-D Lösungen zu rekonstruieren, wird zugleich ein effizienter Bewegungsplaner vorgestellt, welcher auf der Echtzeitintegration eines dynamischen Systems basiert. Infolgedessen erschließt sich der Forschungsbereich zu hybriden dynamischen Systemen für robotisches Fangen. Hierbei ist vor allem der nun mögliche Rückgriff auf existierende Beiträge zum eindimensionalen, prellenden Ball ein Gewinn. Die beispielhafte Anpassung und Erweiterung uniformer, Zeno asymptotischer Stabilität prellender Bälle ermöglicht daraufhin erstmals nachweislich robustes Fangen. Dieser Formalismus wiederum erlaubt das Aufstellen eines quantitativen Stabilitätsmaßes und anhand dessen eine Diskussion der Bedeutung des Relativzustandes zwischen Objekt und robotischem Fangwerkzeug (*Endeffektor*). Die Quantifizierung des Stabilitätsmaßes kann dann in einem weiteren Schritt als Kostenfunktion einer dynamischen Optimierung eingesetzt werden um die Konvergenzgeschwindigkeit bei der Kontaktherstellung zu maximieren. Außerdem erlaubt diese Quantifizierung einen erfolgsorientierten Vergleich verschiedener Fangbewegungen und -systeme.

Fangbewegungen erfordern im Allgemeinen hohe Bewegungsgeschwindigkeiten und Beschleunigungen von ausführenden Robotern. Letztere müssen daher meist nahe ihrer Geschwindigkeits-, Drehmoment- und Leistungsgrenzen betrieben werden. Diese Grenzen andererseits nicht zu überschreiten ist unabdingbar, um die Übertragbarkeit erfolgsorientierter Vergleiche bei der praktischen Umsetzung zu gewährleisten. Dafür schlägt diese Arbeit einen erweiterten kinematischen Formalismus bei nicht-greifender Objektmanipulation für den Fall vor, dass diskontinuierliche Kontakte auftreten. In solchen Szenarien ist der genaue Kontaktpunkt mit einem Endeffektor nicht entscheidend und stellt somit bis zu zwei weitere Freiheitsgrade dar. Diese Freiheitsgrade können der kinematischen Kette eines Manipulators an ihrem Ende als virtuelle, angetriebene Gelenke hinzugefügt werden. Solange kein Objektkontakt besteht unterliegen die neu gewonnenen virtuellen Gelenke keinen Geschwindigkeits- oder Beschleunigungsbeschränkungen. Ein optimierungsbasierter Ansatz nutzt diese Unbeschränktheit beispielhaft um auf-

tretende Beschleunigungsvariationen bei zugleich hohen Bewegungsgeschwindigkeiten entlang einer grob vorgegebenen Bahn zu bewältigen.

In einem letzten Schritt, wird die in ihrer Gesamtheit robuste Lösung robotischen Fangens mit zwei symmetrischen Robotern verifiziert, welche autonom einander Objekte zuwerfen und zuverlässig fangen. Dieses kontrollierbare Robotikszenario erlaubt reproduzierbare experimentelle Ergebnisse, sowie eine statistisch signifikante Anzahl an Wiederholungen pro Systemkonfiguration.

# Contents

# Notations

## Abbreviations

| | |
|---|---|
| AK | augmented kinematics |
| COM | center of mass |
| DOF | degree of freedom |
| P1 | first phase: acceleration of the throwing robot |
| P2 | second phase: acceleration of the catching robot |
| P3 | third phase: flight path tracking of the catching robot |
| P4 | fourth phase: deceleration of the catching robot |
| POC | point of contact |
| SK | standard kinematics |
| SLHS | simple Lagrangian hybrid systems |
| U1 | uncertainty in the initial object state |
| U2 | time-varying uncertainties |
| U3 | uncertainty in the coefficient of restitution |
| UZAS | uniform Zeno asymptotically stable |

## Conventions

### Scalars, Vectors, and Matrices

*Scalars* are denoted by upper and lower case letters in italic type. *Vectors* are denoted by lower case letters in bold italic type, as the vector $\boldsymbol{x}$ is composed of elements $x_i$. *Matrices* are denoted by upper case letters in bold italic type, as the matrix $\boldsymbol{M}$ is composed of elements $M_{ij}$ ($i^{\text{th}}$ row, $j^{\text{th}}$ column).

### Subscripts and Superscripts

| | |
|---|---|
| $\dot{a}$ | first time derivative $\frac{\mathrm{d}}{\mathrm{d}t}a$ (elementwise if a is a matrix) |
| $\ddot{a}$ | second time derivative $\frac{\mathrm{d}^2}{\mathrm{d}t^2}a$ (elementwise if a is a matrix) |
| $\nabla a$ | gradient of function $a$ |
| $a^{-1}$ | inverse of $a$ |
| $\boldsymbol{a}^{\mathrm{T}}$ | transpose of $\boldsymbol{a}$ |
| $a_{\mathrm{r}}$ | $a$ at release time $t = t_{\mathrm{r}}$ |
| $a'$ | $a$ at release time $t = t'$ |

| | |
|---|---|
| $a_\mathrm{O}$ | $a$ of the (flying) object |
| $a_\mathrm{E}$ | $a$ of the robot end-effector |
| $\|\boldsymbol{a}\|_2$ | Euclidean norm of $\boldsymbol{a}$ |

# Variables

## Scalars, Vectors, Constants

| | | |
|---|---|---|
| $a$ | | number of virtually augmented prismatic joints |
| $\boldsymbol{c}$ | | vector of optimization variables |
| $e$ | | coefficient of restitution |
| $e_\mathrm{N}$ | | Poisson's kinetic coefficient of restitution |
| $e_\mathrm{P}$ | | Newton's kinematic coefficient of restitution |
| $\boldsymbol{f}$ | $\mathbb{R}^m \to \mathbb{R}^n$ | map from joint space to task space |
| $F$ | | normal force during impact |
| $g$ | | gravitational constant |
| $\boldsymbol{G}(\boldsymbol{q}_\mathrm{m})$ | $\mathbb{R}^m$ | gravitational vector |
| $h$ | | shortest distance between $\mathcal{S}_\mathrm{E}$ and $\mathcal{S}_\mathrm{O}$ |
| $\boldsymbol{h}$ | $\mathbb{R}^{(3m+a)}$ | generalized constraint vector |
| $H_i$ | | component-wise inverse distance function |
| $\underline{\boldsymbol{h}}, \overline{\boldsymbol{h}}$ | $\mathbb{R}^{(3m+a)}$ | lower and upper bound of generalized constraint vector |
| $k$ | | weighting factor of the null-space projection |
| $K_\mathrm{P}, K_\mathrm{D}$ | | control gains |
| $k$ | | weighting factor of the null-space projection |
| $\boldsymbol{\ell}$ | $\mathbb{R}^m$ | vector of joint lengths |
| $m$ | | number of degrees of freedom in the classic kinematic sense |
| $n$ | | task dimension |
| $p(x_2)$ | | relative acceleration bound depending on sign$(x_2)$ |
| $\boldsymbol{p}$ | $\mathbb{R}^3$ | position of the origin of $\mathcal{T}$ w.r.t. $\mathcal{T}^\mathrm{W}$ |
| $\tilde{\boldsymbol{p}}$ | $\mathbb{R}^4$ | homogeneous representation of $\boldsymbol{p}$ |
| $\bar{\boldsymbol{p}}$ | $\mathbb{R}^2$ | reduced planar representation of $\boldsymbol{p}$ |
| $\boldsymbol{p}^i, \boldsymbol{p}^j$ | $\mathbb{R}^3$ | position in frame $\mathcal{T}^j$, with $i, j \in \{\mathrm{W}, \mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{p}_i, \boldsymbol{p}_j$ | $\mathbb{R}^3$ | position of the origin of $\mathcal{T}^i$ or $\mathcal{T}^j$ w.r.t. $\mathcal{T}^\mathrm{W}$, with $i, j \in \{\mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{p}_{\mathcal{S}_\mathrm{E}}$ | $\mathbb{R}^3$ | position of a point on the end-effector surface w.r.t. $\mathcal{T}^\mathrm{W}$ |
| $\boldsymbol{p}_{\mathcal{S}_\mathrm{O}}$ | $\mathbb{R}^3$ | position of a point on the object surface w.r.t. $\mathcal{T}^\mathrm{W}$ |
| $\boldsymbol{p}_j^i$ | $\mathbb{R}^3$ | position of the origin of $\mathcal{T}^i$ w.r.t. $\mathcal{T}^j$, with $i, j \in \{\mathrm{W}, \mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{p}^\mathrm{H}$ | $\mathbb{R}^{4a}$ | polynomial coefficients of cubic Hermite spline |
| $P$ | | impulse |
| $P_\mathrm{c}$ | | compression impulse |
| $P_\mathrm{r}$ | | restitution impulse |
| $\overline{\boldsymbol{P}}$ | $\mathbb{R}^m$ | upper power limit |

| | | |
|---|---|---|
| $\boldsymbol{q}$ | $\mathbb{R}^{(m+a)}$ | all joints |
| $\boldsymbol{q}_\mathrm{a}$ | $\mathbb{R}^a$ | virtual prismatic joints with $a \in \{1, 2\}$ |
| $\boldsymbol{q}_\mathrm{m}$ | $\mathbb{R}^m$ | classic joints |
| $\dot{\boldsymbol{q}}_\mathcal{N}$ | $\mathbb{R}^m$ | arbitrary joint velocities for projection into nullspace |
| $\underline{\boldsymbol{Q}}, \overline{\boldsymbol{Q}}$ | $\mathbb{R}^{(m+a)}$ | lower and upper bound of joint ranges |
| $r$ | | object circumcircle |
| $t$ | | time |
| $t_0$ | | start of regarded time horizon |
| $t_\mathrm{f}$ | | end of regarded time horizon (duration) |
| $t_\mathrm{r}$ | | release time (throwing) |
| $\underline{\boldsymbol{T}}, \overline{\boldsymbol{T}}$ | $\mathbb{R}^m$ | lower and upper bound of torque limits |
| $u$ | | horizontal object position in Cartesian space |
| $\hat{u}$ | | unit vector horizontal position in Cartesian space |
| $u_\mathrm{E}$ | | horizontal end-effector position in Cartesian space |
| $U_{0,\min}, U_{0,\max}$ | | auxiliary variables for calculating $Z_\max$ |
| $v$ | | horizontal object position in Cartesian space |
| $\hat{v}$ | | unit vector horizontal position in Cartesian space |
| $V$ | $\mathcal{U} \to \mathbb{R}_{\geq 0}$ | Lyapunov function value |
| $V^-$ | | normal velocity before a collision |
| $V^+$ | | normal velocity after a collision |
| $\underline{\boldsymbol{V}}, \overline{\boldsymbol{V}}$ | $\mathbb{R}^m$ | lower and upper bound of joint velocity limits |
| $w$ | | vertical object position in Cartesian space |
| $\hat{w}$ | | unit vector vertical position in Cartesian space |
| $w_\mathrm{E}$ | | horizontal end-effector position in Cartesian space |
| $W$ | | energy part of the Lyapunov function |
| $\boldsymbol{x}$ | $\mathbb{R}^2$ | state vector |
| $\bar{x}_1$ | | maximal remaining rebound height |
| $x_2^\mathrm{rob}$ | | robust initial choice for $x_2(0)$ |
| $\boldsymbol{x}_\mathrm{n}$ | $\mathbb{R}^n$ | $n$-dimensional task vector with typically $n \in \{1, \ldots, 6\}$ |
| $\boldsymbol{x}_\mathrm{a}(0) \ldots \boldsymbol{x}_\mathrm{f}(0)$ | $\mathbb{R}^2$ | test cases for initial relative states |
| $Z$ | | Zeno time |
| $Z_\max$ | | maximal Zeno time |
| $\bar{Z}_\max$ | | uppper bound of maximal Zeno time |
| $\alpha$ | | object flight path angle during ballistic flight |
| $\beta$ | | auxiliary variable for calculating $Z_\max$ |
| $\gamma$ | | relative acceleration between object and end-effector |
| $\gamma^*$ | | desired rel. acceleration between object and end-effector |
| $\gamma_\min$ | | lower uncertainty bound on $\gamma$ |
| $\gamma_\max$ | | upper uncertainty bound on $\gamma$ |
| $\gamma_\mathrm{E}$ | | end-effector acceleration on ballistic flight path |
| $\gamma_\mathrm{O}$ | | object acceleration on ballistic flight path |

| | | |
|---|---|---|
| $\Gamma$ | | flight plane |
| $\delta$ | | auxiliary variable for calculating $Z_{\max}$ |
| $\theta$ | | orientation of the flight plane around $\hat{w}$ |
| $\kappa$ | | relation of acceleration uncertainty bounds and coefficient of restitution (UZAS) |
| $\nu$ | | first time derivative of $h$ (relative velocity) |
| $\nu^-$ | | relative velocity before impact |
| $\nu^+$ | | relative velocity after impact |
| $\nu_{\mathrm{c}}$ | | auxiliary variable for calculating $Z_{\max}$ |
| $\nu_{\mathrm{E}}$ | | end-effector velocity along the ballistic flight path |
| $\nu_{\mathrm{O}}$ | | object velocity along the ballistic flight path |
| $\boldsymbol{\xi}$ | $\mathbb{R}^3$ | state for dynamical system motion planner |
| $\boldsymbol{\xi}_0$ | $\mathbb{R}^3$ | initial state of dynamical system motion planner |
| $\boldsymbol{\xi}_{\mathrm{O}}$ | $\mathbb{R}^3$ | object state in unicycle formulation |
| $\rho$ | | Newton's coefficient of restitution |
| $\sigma$ | | auxiliary variable for calculating $Z_{\max}$ |
| $\boldsymbol{\tau}$ | $\mathbb{R}^m$ | joint torque |
| $\boldsymbol{\omega}$ | $\mathbb{R}^3$ | object rotation with elements $\omega_u$, $\omega_v$, $\omega_w$ |

### Matrices

| | | |
|---|---|---|
| $\boldsymbol{C}$ | $\mathbb{R}^{m\times m}$ | Coriolis matrix |
| $\boldsymbol{J}$ | $\mathbb{R}^{(m+a)\times(m+a)}$ | Jacobian matrix |
| $\boldsymbol{J_W}$ | $\mathbb{R}^{(m+a)\times(m+a)}$ | weighted pseudo-inverse of the Jacobian matrix |
| $\boldsymbol{M}$ | $\mathbb{R}^{m\times m}$ | inertia matrix |
| $\boldsymbol{P}$ | $\mathbb{R}^{m\times m}$ | projection matrix (nullspace) |
| $\boldsymbol{R}$ | $\mathbb{R}^{3\times 3}$ | orthonormal rotation |
| $\boldsymbol{R}_i$ | $\mathbb{R}^{3\times 3}$ | rotation of $\mathcal{T}^i$ w.r.t. $\mathcal{T}^{\mathrm{W}}$, with $i \in \{\mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{R}^i_j$ | $\mathbb{R}^{3\times 3}$ | rotation of $\mathcal{T}^j$ w.r.t. $\mathcal{T}^i$, with $i, j \in \{\mathrm{W}, \mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{T}$ | $\mathbb{R}^{4\times 4}$ | homogeneous transformation |
| $\boldsymbol{T}^i_j$ | $\mathbb{R}^{4\times 4}$ | configuration of $\mathcal{T}^j$ w.r.t. $\mathcal{T}^{\mathrm{i}}$, with $i, j \in \{\mathrm{W}, \mathrm{E}, \mathrm{O}\}$ |
| $\boldsymbol{W}$ | $\mathbb{R}^{(m+a)\times(m+a)}$ | weighting matrix for the pseudo-inverse |

## Sets

| | |
|---|---|
| $\mathcal{A}$ | compact set of equilibria |
| $\mathcal{A}_{\mathcal{Z}}$ | Zeno equilibria |
| $\mathcal{C}$ | subset of the Euclidean space describing the continuous domain |
| $\mathcal{D}$ | subset of the Euclidean space describing the discrete domain |
| $\mathcal{F}$ | set-valued mapping describing the continuous dynamics |
| $\mathcal{G}$ | set-valued mapping describing the discrete dynamics |

| | |
|---|---|
| $\mathcal{H}$ | hybrid (bouncing ball) system |
| $\mathcal{N}$ | nullspace of a matrix |
| $\mathcal{P}$ | constants defining a flight path or trajectory |
| $\mathbb{R}$ | natural numbers |
| $\mathcal{S}_{\mathrm{E}}$ | end-effector surface |
| $\mathcal{S}_{\mathrm{O}}$ | object surface |
| $\mathcal{T}$ | coordinate frame |
| $\mathcal{T}^{\mathrm{W}}$ | world coordinate frame |
| $\mathcal{T}^{\mathrm{E}}$ | end-effector coordinate frame |
| $\mathcal{T}^{\mathrm{O}}$ | object coordinate frame |
| $\mathcal{U}$ | Lyapunov function input space |
| $\mathcal{X}_{\bar{Z}_{\max}}$ | subspace of $\mathcal{C} \cup \mathcal{D}$, where $Z_{\max} < \bar{Z}_{\max}$ |

# 1 Introduction

Robots are preprogrammed machines that are capable to autonomously perform a set of physical tasks based on predefined inputs. In their most prevalent role, which is the manipulation of objects or parts, robots are equivalently referred to as *manipulators*. Typical goals of manipulation are to change the state (e.g., position, orientation, or velocity) of objects or to alter these objects (e.g., assembly, drilling, or welding). The decision whether a robotic solution is favored over its human counterpart is mostly driven by economical considerations. Beyond the per unit costs during a robot's life-cycle, such considerations may also include hazard potential, range of skills, manipulation precision and repeatability, reliability, or continuity and speed of throughput per workstation. Amongst these potentially relevant considerations, manipulation speed (throughput) has one of the largest leverage effects. Moreover, as speed of available manipulation solutions increases, the consequences of various types of uncertainties can rise from negligible up to critical, which imposes new challenges to researchers and applicants.

Today, a variety of commercial robotic systems outperforming humans in a subset of their skills is available and continuously growing in number. The particular progress robots have made in major human skill categories, however, differs substantially. For example, endurance, strength, or precision can be directly scaled by using hardware components of better quality, a larger size, or with more power. Such improvements are mostly a matter of expense and component availability. Many of these components have evolved over decades before their deployment to complex robotic systems. The interaction of already widely deployed robotic systems with the environment typically happens in highly controlled scenarios in which the states of all participants, both manipulator and manipulated, are known with high precision. Uncertainties are minimized in advance whenever possible. Conversely, humans often outperform robots in terms of dexterity or skill acquisition in dynamic and uncertain environments. Here, the choice of hardware components plays an inferior role. Instead, completely new and highly specialized methods developed from a deep understanding of a task or process are needed to facilitate a robot's fast, efficient and robust interaction with the environment.

The most prominent example regarding the number of deployed robots is automotive production. Without the need to rest, robots can operate almost constantly, do not suffer from sickness or go on strike and, therefore, outperform humans even if they output fewer units per time. Robots can lift and turn entire vehicle bodies or assume weight-lifting duties when workers mount tires to a car. Precision can be scaled to

high absolute and relative accuracy benefiting machines from surgery robots [5, 30] to welding robots. Alternatively, specialized feeding systems can enforce the pose of manipulated objects. If sensing is required, ambient conditions can be adapted to the sensor requirements. Moreover, objects can be beneficially placed relative to the sensor, as the task is known in advance [44]. With high absolute accuracy comes high repeatability and thus, errors become reproducible and can be eliminated permanently and for all identical stations at once. In comparison, human workers must be trained individually without guarantees of improvement. Arguably, humans increase uncertainty with most of their actions but are naturally capable of compensating for uncertainties quickly with subconscious adaptations to changing situations. For example, in household tasks where objects vary in pose or are hidden behind several other objects in a cabinet. Preparing a meal with collaborators requires reactive decisions of the next action from a large pool of yet unsolved skills ranging from peeling or cutting fruit and vegetables to making a pizza dough [80].

In uncertain scenarios, the shape, position, velocity and acceleration of manipulated objects is, at most, approximately known in advance. Therefore, the future object motion is only partially predictable. The main consequences of such uncertain dynamics in robotic manipulation are unexpected, although nonetheless inevitable, collisions with objects or the environment. Manipulation solutions with precisely planned contact points at a particular instant are, therefore, not applicable. To teach robots new skills or tasks, a significant body of research devotes itself to the reduction of environmental uncertainty by proposing faster and more accurate perception methods. Some solutions must even employ computing clusters to receive timely predictions and motion plans [10, 11]. Still, some uncertainty always remains and must be solved by robust task plans that compensate for collisions and uncertain contact points, states and times.

Given the inevitability of collisions during uncertain manipulation, a mixture of continuous (no contact) and discrete (at collisions) dynamics applies. Hence, a framework suited to describe and control these different dynamics becomes necessary. Hybrid dynamical systems [18, 34], intensively studied in the control literature, constitute a suitable, albeit still evolving [62], framework. The application of hybrid control solutions with the potential to be useful for solving manipulation problems is an engineering challenge in itself. Many approaches, such as for impacting systems, are tailored to linear problems of low dimensionality [64]. Other approaches build on assumptions like constant acceleration [35], which does not hold in dynamic environments with moving reference frames, for example, using a robot to catch an object by adapting the robot's motion to the object in flight.

After identifying a robust task plan in theory and simulation, the approach is deployed in robotic systems. In this step, the displacement and motion of a series of robot joints, with a manipulation tool at its end (*end-effector*), is planned to perform the task. For tasks of low velocity and dexterity requirements, this step is inexpensive as textbooks provide many real-time capable motion planners [88, 90]. With increasing dynamic requirements, this step becomes more challenging. Two general approaches

may be distinguished: offline and online motion planning. The advantage of offline motion planning is the inferior role of a method's computational complexity. If the task is sufficiently known in advance, an almost globally optimal robot motion can be planned, which allows for efficient exploitation of the available hardware. Or, the offline planning process can include the right choice of hardware. Dynamically challenging tasks such as softly catching objects may even necessitate offline planning as a basis. An immediate reaction to unexpected scenario changes and deviations is not naturally envisaged. However, sophisticated and robust task plans may compensate for a sufficient range of deviations to make offline motion planning sufficient. In contrast, online motion planning is tailored to react to deviations and disturbances during operation. These methods typically plan the robot motion for a short time horizon or single control cycle, including the time for perception and interpretation of the current robot and environment state. Solutions are, therefore, at most locally optimal and can run the robot into its constraints even though a feasible global solution likely existed. As a result, methods attempting to unite the advantages of both approaches have been proposed [66, 83].

## 1.1 Challenges and Solutions to Robotic Object Manipulation

Due to its pervasiveness in robotics and beyond, the term *manipulation* has various definitions [59]. This thesis applies the 2015 NASA definition: "Manipulation pertains to making an intentional change in the environment or to objects that are being manipulated" [61, p. 13]. A robotics or control engineer attempting to teach a robot manipulating objects might be able to become more precise, saying 'to change object states through selective contact', i.e., to change an object's position, orientation, velocity, or rotation. Picking up and placing an object is a widely deployed example of this definition. The machining of objects is not part of the manipulation definition in this thesis but is sometimes used by engineers with other manipulation goals, which can include but are not limited to welding, bending, drilling, cutting, and milling.

Various robotic systems solving basic manipulations tasks are already commercially available. Nevertheless, a majority of tasks from production to everyday life still require humans, who can handle unstructured situations naturally. The engineering solution for handling similarly unstructured tasks are specialized robotic systems. The design of these systems is a diverse challenge as expertise in various research fields is necessary. One has to either select amongst various available methods and adapt or extend these approaches. Or, one has to develop new methods. In this thesis, the following fields are distinguished, whereas the requirements for each field scale with task complexity: system architecture design, perception and prediction, task modeling and planning, robot motion planning, robot control, and robot design. The relative importance of

each field depends on the task. For example, autonomous household robots typically face cluttered environments, demanding sophisticated perception, interpretation, and navigation [44]. In physical human-robot interaction, the focus beyond sensing is also on robot control to provide compliant and safe interaction [4, 25]. Robots that catch objects are rather designed with stiff controllers for accurate trajectory tracking at high velocities [83]. Moreover, catching causes inevitable collisions that must be considered during task planning [84]. The following narrative provides a brief review of the fields as distinguished in this thesis by highlighting a selection of goals, challenges, and exemplar solutions.

**System design** for a single or set of tasks starts out most developing procedures and typically requires several iterations. System design is devoted to the choice of software and hardware architecture, including the definition of interfaces and interconnections. At times, system design includes the environment in which a robot acts, for example, when obstacles are removed or feeders provide objects in a predefined orientation. Ideally, design iterations converge to a system architecture that decouples the manipulation problem into well-separated subproblems that can be individually solved. Therefore, system architecture design is a crucial step as it influences the resolvability of all subsystems and subproblems.

**Environment perception and prediction** constitutes the first component in the event chain and the basis for all subsystems described hereafter. Perception includes sensor data processing ranging from force feedback [27, 68] to vision, both of which are subject to quantization errors and noise. Hence, solutions in hardware and software aim to keep uncertainty to a minimum. Scene interpretation is also part of the perception procedure, which includes, for example, the classification and state estimation of objects. Contributions in computational learning techniques, driven by increasing computational power, have advanced the field of perception in recent years [44, 46]. A major drawback of most learning-based solutions, however, is the absence of guarantees regarding successful classification. Therefore, the reliability of information has become an important challenge in perception [58]. As a result, solutions that depend on such uncertain identification techniques typically face problems during failure analysis as it is hard to determine whether a failure is due to perception problems or insufficient task planning [46, 78].

In dynamic environments, the perception challenge extends to the prediction of future environment states of, for example, moving or flying objects, walking humans, or other moving obstacles. The time available for perception and prediction is often dictated by the scenario at hand, for example, the time until a fast flying object reaches a robot's workspace or until a collision with an obstacle is inevitable. In such cases, computational complexity becomes critical and demands highly efficient detection and prediction methods, additional computational power, or reduced task speed. Therefore, attempts to solve the complex challenge of perception and prediction sometimes mask underlying questions in related fields that should instead be answered to provide robust manipulation solutions. Against this backdrop, this thesis has a strong focus on the

associated domain of task and robot motion planning. Hence, in the present research, a throwing robot replaces the typically highly complex perception unit necessary to quickly perceive and catch flying objects with bounded uncertainty. Moreover, such a throwing robot allows the intentional introduction of additional uncertainty to verify the explicitness of previously claimed robustness constraints.

**Task modeling and planning** in the context of manipulation refers to planning the motion of the manipulator tool to interact with the environment or object to achieve desired changes. This tool can be of any kind, from a screwdriver to a hammer or from a humanoid hand to a simple plate, and is typically mounted at the end of a robot's kinematic chain. Therefore, the manipulator tool is commonly referred to as an *end-effector*. Achieving the desired change in a robust manner can be a challenging task as information from the perception step is fraught with uncertainty, at least to some extent. A manipulation goal definition and resulting manipulation plan must explicitly consider the potential magnitude of these uncertainties to guarantee successful task execution. Otherwise, task failures can occur, for example, caused by neglected or overlooked collisions [78], which cannot be conclusively explained.

End-effectors similar to human hands are a common choice as they can enable robust grasping by form or force closure. The grasp can compensate minor pose estimation errors and makes the object state precisely controllable. Opening and closing a gripper is, however, a time-consuming process that cannot be arbitrarily accelerated while keeping the uncertainty compensation constant. For example, the closure time of a gripper depends on the distance between the fingers and the object, whereas this distance also represents the potential range of pose uncertainty compensation. The faster and the more dexterous a manipulation task becomes, the more challenging grasping becomes.

Dynamic nonprehensile manipulation [55] explicitly addresses such dexterous and fast tasks by exploiting, rather than merely tolerating, task dynamics [60]. End-effectors in nonprehensile manipulation are typically generic, for example, a plate [39, 98] or a box [84], and can be used for a range of tasks and objects [74]. The identification and modeling of relevant task dynamics is much more challenging, though, whereas solutions are potentially more robust and generic with learnings that can also advance grasping manipulation.

**Robot motion planning** aims to generate robot motions at the joint level that cause the end-effector to move as desired by the task planning solution. This task-level approach also defines the operational requirements of the joint motion planner: offline, online, or even real-time. For example, offline planning can be sufficient if the environment is controllable and the task planner is robust [84]. Online approaches enable reacting to the current environmental situation and are, therefore, more flexible and extend the range of manipulable objects and object states. Real-time capable planners are even faster and typically consist of local, analytic inverse kinematic solutions, which are computationally inexpensive. These planners can adapt to changing task requirements in every time step.

Particularly challenging in kinematic trajectory planning is the presence of kinematic and dynamic constraints that limit, for example, joint displacement range, joint velocity, motor torque, and motor power. With the increasing speed and dexterity requirements of a task, the robot must be operated closer to these limits. Standard online and real-time planners, in particular, do not always identify a feasible ad-hoc solution even though one is likely to exist when using a global solution mechanism offline. Depending on the method, these fast but local planners usually reduce task speed, deviate from the desired path, or violate constraints. Tasks that do not allow for such relaxations, but still demand fast and precise end-effector motions at the edge of a robot's dynamic capabilities, must provide other means of scalability, such as the degree of robustness or task duration, which are addressed in this thesis. Conversely, kinematic trajectory planners can be adapted to specific tasks and, for example, initialized with approximate task knowledge offline to provide close to optimal robot motions online or in real-time [66, 83].

**Robot motion control** aims to track the desired joint motions generated in the previous step. Motion controller candidates typically face noisy measurement data from various types of transducers, which should be compensated. Moreover, the accuracy requirements for kinematic trajectory tracking can vary during a task. For example, position accuracy in pick and place tasks may only need to be high during environment interaction (pick or place action), whereas path deviations in between might be acceptable.

The range of solutions is broad and well developed [88, 90]. Some robot motion planners have the controller integrated or working at the position level (operational space) such that motion planning and control reverse order in the event chain [45]. In this thesis, the order is as described in this section and the joint controller is entirely decoupled. Moreover, the motion controller is assumed to be provided with feasible joint trajectories that do not violate any constraints while the trajectory is accurately tracked.

**Manipulator design** strongly influences the range of tasks a robot can potentially solve. The kinematic and dynamic model, with its respective constraints, varies depending on the number, class (rotating or linear) and interconnection of joints (serial or parallel), length and material of the joint connectors (*links*), or the size and transmission of joint motors and gears. Manipulator design is, however, often not considered variable when designing manipulation solutions because the use of available hardware is a precondition, especially in research laboratories. Instead, for example, optimization-based task and motion planning methods are used to maximally exploit robot capabilities. If a robot's capability is still insufficient for handling a particular task, advanced manipulation techniques from the field of dynamic and nonprehensile manipulation can be adapted and enhanced [55, 60, 74, 83].

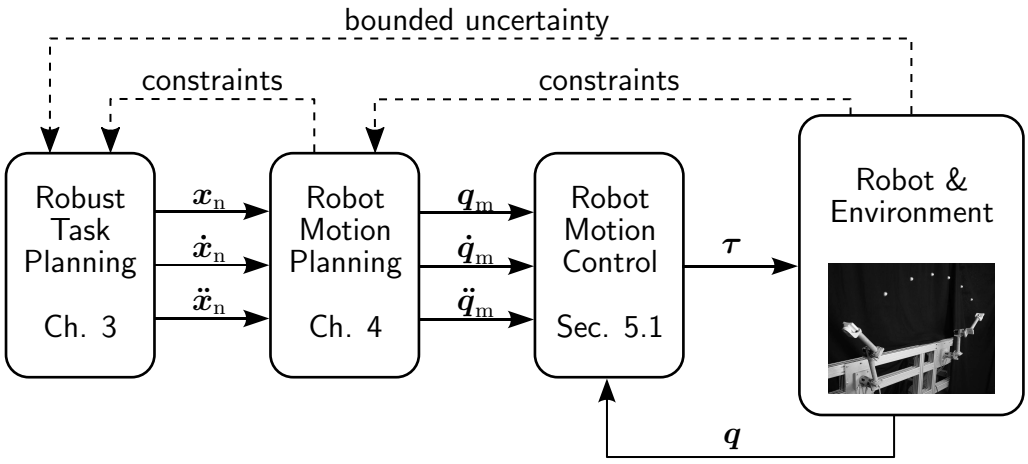


**Figure 1.1:** System design for robot-robot throwing and catching based on robust open-loop task planning and dynamically feasible robot motion planning.

## 1.2 Contributions

This thesis aims to expand the boundaries of the research field examining fast and dexterous environment interaction. More precisely, the focus is on establishing contacts in a fast and robust manner during robotic manipulation tasks. For this purpose, both the manipulator and objects are considered to move with high absolute velocities in the moment of contact. The critical role of scenario uncertainties is explicitly considered in the modeling process, for example, uncertain object states or the inevitable occurrence of collisions, which are particularly known for their highly state-sensitive prediction models. Against this background, the task of catching a fast flying object without grasping serves as an illustrative and challenging example throughout this work.

For the particular problem of catching fast flying objects, the robotic manipulation system is designed in the context of the fields introduced in Section 1.1, and Figure 1.1 depicts a schematic overview of the system. Compared to most other catching systems reported in the literature, this system lacks a component typical of such manipulation tasks: perception using visual feedback. Instead, a throwing robot provides repeatable flight trajectories. Similar to a visual perception system such as in [46], a throwing robot introduces bounded, but undetectable, uncertainty. Moreover, robotic throwing enables the intentional introduction of additional uncertainty, which helps to verify the robustness bounds claimed in theory. Besides this uncertainty, the ballistic flight trajectories cast by the throwing robot can be assumed to be similar. Therefore, the research focus in this thesis is shifted from the perception and prediction problem to the problem of fast object manipulation under uncertainty. As a result, sufficiently robust task and robot motion planning allow for cyclic open-loop throwing and catching. With this goal in mind, the remainder of the thesis contributes to the existing body of works the following modeling approaches and methods:

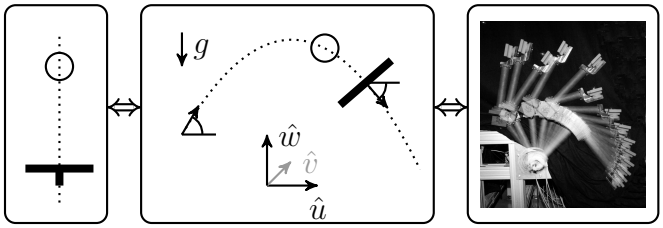


**Figure 1.2:** A dynamical system parametrization enables ballistic nonprehensile catching with dynamically feasible offline motions using hybrid bouncing ball formalisms.

1. **Generalized problem definition:** A problem definition is formulated for catching (fast) flying objects of arbitrary shape and is independent of the robotic problem. In contrast to goal formulations in previous works defined as those which exactly match the object state with a robot's end-effector motion [78], in this work, the formulation explicitly allows for provable uncertainty compensation, i.e., robustness:

   (i) The goal of catching arbitrarily shaped objects is defined by the shortest distance and its time derivative between the object and catching device.

   (ii) The large number of uncertainty sources in practical catching is reduced to three representative uncertainty types, which must be compensated for to achieve provably robust catching: (U1) uncertain initial relative states, (U2) uncertain relative accelerations over time and (U3) uncertain state resets at collisions.

   *The problem and uncertainty definitions were first published in the IEEE Transactions on Robotics article [84].*

2. **Hybrid system stabilization for ballistic robotic catching:** Ballistic robotic catching is connected to hybrid control theory to enable convenient modeling and stability analysis of the inevitable occurrence of collisions. Uncertainties, which are mostly due to the limited knowledge of the object state, make ideal contact transitions from *no contact* to *continuous contact* unlikely if both the object and end-effector are rigid. Hence, collisions occur inevitably, a topic which has been neglected in previous works.

   (i) The gap between hybrid control theory and ballistic robotic catching is closed with a dimensionality reduction. Under the assumption of ballistic free flight, the occurring sequence of collisions can be parametrized as a one-dimensional hybrid bouncing ball problem along the flight trajectory. A real-time capable, task motion planner is proposed to reconstruct the desired, full-dimensional end-effector motion. As a result, robust solutions for the hybrid bouncing ball, well-studied in hybrid dynamical systems theory, can be adapted and extended for object catching.

(ii) As a consequence of the previous contribution (i), stability notions for the one-dimensional set-valued bouncing ball problem are extended for object catching. At first, robustness against an uncertain kinematic coefficient of restitution is proven for a range of object states and relative accelerations. Second, an analytical solution for the initial relative velocity is derived, which maximizes distance uncertainty compensation and, therefore, significantly increases catching robustness. Third, an analytic prediction of the maximal remaining rebound height for uncertain bouncing balls is derived, which enables quantification of catching robustness with box-shaped end-effectors of different heights.

(iii) Simulation studies have revealed that the proposed methods can compensate for the significant offset in the relative acceleration. A full-dimensional simulation with a rotating cube object demonstrates that this compensation includes deviations neglected during dimensionality reduction.

*The above contributions have been published in the IEEE Transactions on Robotics article [84].*

3. **Feasible kinematic trajectories for manipulation with intermittent contacts:** Inspired by humans, who only approximately plan contact with objects, the free choice of contact is formally introduced as two virtual joints at the end of the robot's kinematic chain. The major advantage of these two joints is their unconstrained nature regarding velocity and acceleration. As object catching is a highly dynamic task that demands potential catching robots operate close to their limitations, these virtual joints can be a key to dynamically feasible robot motions.

   (i) A first approach parametrizes the additional redundancies gained with the virtual joints offline. Constrained optimization then exploits these joints to maximize task robustness while guaranteeing kinematic and dynamic feasibility. Adaptations to online changes are not possible with this approach, but experiments described in Chapter 5 reveal that the robustness is still sufficient for reliable task execution.

   (ii) A second approach formalizes the choice of weighting parameters for inverse differential kinematic trajectory planners. Two of which are compared against each other and the kinematic formulation without the novel virtual joints.

   *The first approach is part of the journal article [84] published in the IEEE Transactions on Robotics, whereas the second approach has been published in the IEEE Robotics and Automation Letters with [83].*

4. **Experimental demonstration of robust catching:** A simple robotic system consisting of two robots, each with only two actuated joints, is used as a testbed to

demonstrate the catching robustness achieved with the aforementioned contributions. The experimental setup is distinctive due to the absence of visual feedback or grasping devices. Extensive experiments have demonstrated that both are not necessary components for reliable catching in the presence of uncertain object states induced by a throwing robot. These experiments have also suggested that the above methods might even allow simultaneous catching of multiple objects with different shapes, such as spherical and polygonal.
*The experimental material has been partially published in the IEEE Transactions on Robotics article [84] and in the IEEE Robotics and Automation Letters article [83]. Both articles have open-access videos available on the publisher's homepage.*

## 1.3 Thesis Outline

After the introduction, the problem of catching fast flying objects is formulated in Chapter 2, which includes an extensive review of the relevant dynamics. Here, how the problem can be decoupled more efficiently compared to previous work is also discussed.

In Chapter 3, the inevitable occurrence of collisions is regarded at the task level and, therefore, treated as independent from the robot motion planning challenge. Dimensionality reduction through parametrization of the flight trajectory and efficient task planning close the gap between ballistic catching and the hybrid bouncing ball problem well-known in the hybrid control literature. Extensions proof and quantify robustness regarding the uncertainty types defined in the previous chapter. Simulations at the end of the chapter illustrate the contributions of this thesis to the wider research field.

Chapter 4 details the robot motion planning challenge imposed by the high operation velocities in robotic catching. A novel kinematic notion enables formal exploitation of the free choice of contact to achieve dynamically feasible robot motions offline and in real-time.

Chapter 5 demonstrates, using the example of a robot-robot experiment, the significance of the contributions provided by this thesis.

Chapter 6 then provides a review of the achievements of the thesis and discusses ideas for future extensions.

# 2 The Problem of Catching Fast Flying Objects

Catching fast flying objects is a complex challenge, as detailed in Chapter 1. In this chapter, the problem of softly catching fast flying objects is divided into two conceptual dimensions to enable clear references and focused discussions. In the first conceptual dimension, two problem levels are distinguished, which are related to two physically different challenges: task motion planning and robot motion planning. The second dimension divides throwing and catching into four temporal phases characterized by different goals and challenges at both levels. Then, the necessary mathematical and physical background for both problem levels are reviewed. A novel generalized formulation of the catching problem detailed in Section 2.3.3 and integration of various uncertainties into three representative uncertainty types in Section 2.3.5 constitute a research contribution, which is also presented in [84].

## 2.1 Static Catching and Dynamic Catching

Robots catching an object is a challenging and frequently considered testbed to demonstrate the performance of object tracking combined with motion planning in highly dynamic environments. A successful catch relies on effective solutions in both tracking and planning. Inversely, shortcomings of the proposed solutions, such as neglected dynamic feasibility or inaccurate estimation of the object state, typically account for failed attempts. The complexity of many proposed systems hinders an analytic derivation of the mixture of errors that lead to the observed failures. This research discusses the inevitable occurrence of collisions as a critical, but often overlooked, shortcoming of previous works.

Works focusing on the visual tracking and prediction problem before the actual catch often implement *static catching* (*hard catching*) with a gripper [11, 46, 47, 52, 80], where the goal is to track and predict the object flight trajectory with high precision and as quickly as possible. The remaining time is then used to move the catching robot to a reachable kinematic goal posture such that the gripper intercepts the flight path and catches the object at the right moment [24]. This approach generates impact forces and a rebound that can cause the object to bounce off the gripper in cases of uncertainty. Thus, static approaches are limited to soft, relatively light objects or compliant end-effectors to prevent permanent damage. For example, in [26], static catching was used as a challenge for visual servoing with a stereo vision system in which the robot

motion planning was based purely on visual information. At the end of the kinematic chain, the robot is equipped with a bucket to catch a ball. In another approach, online optimized intercepting motions at a kinematic level allowed for faster reactions and, therefore, shorter flight distances [11]. A recent and significant leap forward in tracking and predicting flight of objects was made in [46] where programming-by-demonstration accurately predicted flight trajectories, including rotations for objects of complex shapes. Moreover, the approach could predict the best static catching pose for the gripper. In these static and grasping-based approaches, the contact time estimation was crucial to close the gripper at the right time. Nevertheless, the focus on kinematic interception leads to high impact forces if both the manipulator and object are not specifically designed.

*Dynamic catching* (*soft catching*), in which the robot adapts to the motion of the object [9, 41, 42, 85, 93], enables handling a wider range of objects and is less sensitive to timing inaccuracies [78]. Without a gripper, the goal is formulated as a generic non-prehensile [55, 74] catching task based on the fundamental dynamics [81] that model continuous contact between a free-flying object and the robot end-effector. A very simple nonprehensile robot-robot (i.e., with only 1-DOF each) scenario for spherical objects was simulated in [92] under the assumption of collision-free contact establishment.

Interestingly, one of the first catching systems accomplished the task *dynamically* (velocity and acceleration matching) [41, 42] and not statically. In this early work, a 4-DOF robot with a gripper closed perpendicular to the flight direction to avoid object impacts with the end-effector. The dynamic approach was necessary because vision systems were still too imprecise to determine accurate arrival times. The grasping end-effector also took a considerably long time to close. at about the same time, [17] proposed a "mirror law" to generate the motion for a dynamic catch in combination with a well-tuned PD-controller after the initial impact. Later, a single camera in a robot's hand was sufficient for successful interception [22, 52]. Here, the flight path was followed to dissipate impact energy while the gripper closed. The related problem of kinematic and dynamic feasibility was not formally treated in these works, except for [48] in which static and dynamic catching were compared and robot dynamics received attention.

More recent work has addressed some of the *feasibility* problems in dynamic catching. In [9], the goal definition was extended to match the acceleration of a thrown basketball. In cases of kinematically or dynamically infeasible desired robot motions, an indirect catch is introduced, i.e., a single controlled rebound before the actual catch. In [67], the idea of direct and indirect catching was generalized to polygonal objects. Kinematically feasible offline trajectories were generated in [78] as the basis for a Linear Parameter Varying (LPV) approach. Asymptotic convergence to the object's trajectory was demonstrated with the LPV system definition. However, dynamic feasibility was neglected, which was later considered responsible for some of the failed catching attempts in the experimental evaluation.

## 2.2 Dynamic Throwing and Catching

Majority of research related to catching objects with robots considers human throwers, as reported in Section 2.1. Having a human involved for such a challenging but everyday task is a motivating and attractive idea. A second glance, however, reveals disadvantages from a scientific perspective, as experimental evaluation strongly correlates with the particular individual. Of note, when proposed methods succeed at a rate of less than 100%, identifying reasons for failures is difficult and often limited to educated guessing. Reproducible failure scenarios, with a significant amount of trials, involving particular (relative) object states or robot motions, are rarely reported. Only the trivial case of an object flight that does not pass through a robot's workspace sufficiently centrally is frequently mentioned. In combination with neglected robot dynamics and constraints, the applicability of the most often proposed methods for one's own robot is difficult to judge given the provided data, even for experienced roboticists.

As a consequence of the limitations imposed by uncertain human throwing, this research uses a controllable robot-robot scenario to enable reproducible and statistically significant results. Moreover, throwing and catching are performed *nonprehensile* and *dynamic*. The nonprehensile approach combined with rigid objects and end-effectors is sensitive regarding impacts, which in turn, are inevitable and sensitive regarding state uncertainties of flying objects. Moreover, the absence of grasping requires a dynamic manipulation approach, which refers to the active consideration of the dynamics of the manipulated (here, flying) object in all planning stages. For example, during nonprehensile throwing, the robot motion must be planned such that the object has a known position on the end-effector at the point of release. As a result, the effect of uncertain object states and consequent impacts become visible and repeatable, as these collisions are occasionally not compensated for by a gripper. From a user perspective, the dynamic approach, which includes the robot end-effector adapting to the flight motion, provides supplemental advantages because potential impact forces remain low, preventing object damage. Furthermore, the potential interaction duration during catching increases from a few milliseconds, which is typical for static catching, to a few hundred, allowing the development of control strategies for the catch.

Hereafter, the problem of robot-robot throwing and catching is discussed along with two conceptual dimensions. Two problem levels are distinguished in the first dimension concerning two physically different challenges: task motion planning and robot motion planning. Section 2.2.1 briefly characterizes both levels. The second dimension divides throwing and catching into four temporal phases characterized by different goals and challenges at both levels. Section 2.2.2 briefly characterizes these four phases (P1-P4).

### 2.2.1 The Two Level Problem: Task Motion and Robot Motion

Catching a thrown object demands a combination of several research fields in automatic control and robotics. The challenges range from detecting the flying object to planning

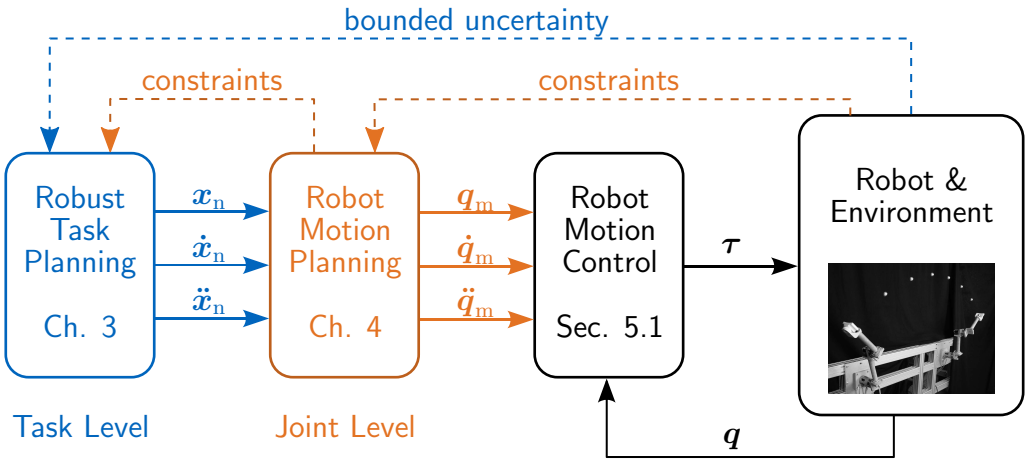


**Figure 2.1:** System design for robot-robot throwing and catching. On task level, a robust open-loop motion for the end-effector is planned. On joint level, dynamically feasible robot motions are generated to move the end-effector according to the task plan.

the interaction for successful catching or performing the interaction with a real robot. As this research resolves the perception challenge with a controllable environment by using a throwing robot, planning the catching interaction and the robot motion are the major subjects, which are distinguished as follows (see Figure 2.1 for an illustration):

- **Task Level** refers to the challenge of identifying an interaction with free flying objects that leads to robust catching, independent of the catching agent[1]. *Robust*, in this context, refers to the ability of repeatedly compensating for a significant number of uncertainties in the environment (e.g., an uncertain object state) and a set of environment parameters (e.g., object shape or material). The interaction is performed with a catching device constituting the *task system* together with the flying objects. The catching device can be, for example, a plate, box, or gripper and is the only part of the task system that can be accelerated by the catching agent, making it the input of the task system.

- **Joint Level** in the context of robust catching refers to a robot agent and the challenge of planning a physically feasible motion for every joint of the robot. All joint motions that might become necessary during operation must be guaranteed to be feasible to achieve reliable task execution. At the joint level, feasibility refers to staying within the limits of the joint range, joint velocity, and joint peak torques. Moreover, it is characteristic for dynamic catching motions that high velocities are combined with complex trajectories requiring operation close to the defined limits. As a result, motor power limits must also be considered. Joint-level planning can be based on different degrees of offline knowledge ranging from

[1] not necessarily a robot from the task level perspective

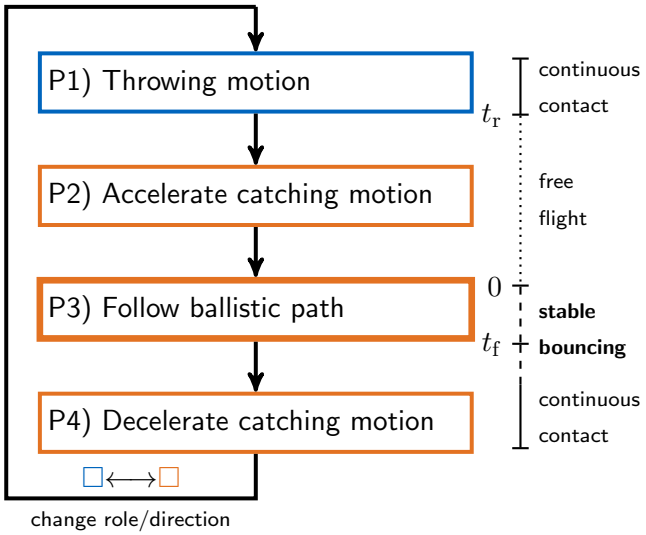


**Figure 2.2:** Block diagram outlining the four phases for joint robot-robot throwing and catching.

a completely known task plan to only approximate task knowledge. In the latter case, the task plan adapts to updated environment knowledge within certain bounds, enabling closed-loop catching if task and joint planners are real-time capable.

### 2.2.2 Four Phase Approach: Throwing, Acceleration, Tracking, Deceleration

For a focused discussion, this research distinguishes between four general robot motion phases per each throwing and catching sequence, as visualized in Figure 2.2:

P1) A first robot throws an object. In this research, optimal control based motion planning maintains dynamic feasibility and ensures dynamic fixation of the object during acceleration. At nonprehensile release, limited and repeatable uncertainty in the object state remains.

P2) A second robot accelerates from rest to the goal state, which is the initial state of P3. Again, optimal control based motion planning is used to solve the boundary value problem in the presence of kinematic and dynamic constraints.

P3) The second robot performs a decelerating motion on the ballistic flight path. Dynamic requirements that enable robust ballistic catching (Chapter 3) and feasible motion planning (Chapter 4) are the major subjects of this research. Along with the discussion, special attention is paid to the relative state between object and end-effector at the start of the following motion.

P4) The second robot must leave the ballistic flight path at some point while still in motion. Again, optimal control based motion planning is used, but here, negative relative acceleration in normal end-effector direction must be applied. Due to kinematic and dynamic limitations of typical robots, a continuous contact at the end of P3 is mostly not possible or not desirable from an efficiency (throughput) perspective.

## 2.3 Task Motion Planning Problem

This section introduces fundamental models describing the task system. Section 2.3.1 introduces the relevant coordinate systems and their generalized relationships. Section 2.3.2 defines the free-flight dynamics of rigid objects, for which Section 2.3.3 then proposes a generalized problem definition. Section 2.3.4 discusses models for the inevitable occurrences of collisions. Finally, Section 2.3.5 formulates three representative types of uncertainties, which aggregate the majority of the versatile uncertainties in reality.

### 2.3.1 Homogeneous Transformations

In this section, common fundamental scenario descriptors [56, 88] are introduced on task level and independent of a particular robotic kinematic structure.

Consider general rigid bodies, each with an orthogonal frame $\mathcal{T}$ attached to it. The configuration of $\mathcal{T}$ with respect to a world frame $\mathcal{T}^{\mathrm{W}}$ is represented by the matrix $\boldsymbol{T} \in \mathbf{SE}(3)$ defined as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}, \tag{2.1}$$

whereas $\boldsymbol{R} \in \mathbf{SO}(3) \in \mathbb{R}^{3\times3}$ is an orthonormal rotation matrix and $\boldsymbol{p} \in \mathbb{R}^3$ is the position of the origin of $\mathcal{T}$ in $\mathcal{T}^{\mathrm{W}}$. The particular attachment of $\mathcal{T}$ to its respective body may vary subject to the task need.

Two frame types beyond $\mathcal{T}^{\mathrm{W}}$ are considered in this research: object frames $\mathcal{T}^{\mathrm{O}}$ and an end-effector frame $\mathcal{T}^{\mathrm{E}}$. The origin of object frames $\mathcal{T}^{\mathrm{O}}$ is attached to the object's center of mass and the axes are aligned with the object's principal axes of inertia. The end-effector frame $\mathcal{T}^{\mathrm{E}}$ has the origin fixed on the end-effector surface, the first two axes are parallel to the surface and the third axis is normal to the end-effector surface. Ideally, the end-effector is later attached to the robot such that one of the first two axes is also parallel to the last joint's axis of rotation while the other axis is perpendicular to it.

Furthermore, the configuration of a frame $\mathcal{T}^j$ with respect to another frame $\mathcal{T}^{\mathrm{i}}$ is denoted by

$$\boldsymbol{T}^i_j = \begin{bmatrix} \boldsymbol{R}^i_j & \boldsymbol{p}^i_j \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}, \tag{2.2}$$

whereas $i, j \in \{\mathrm{W}, \mathrm{E}, \mathrm{O}\}$ correspond to the world frame, robot's end-effector frame and object frame, respectively. In case of $i = \mathrm{W}$ descriptors are omitted in the remainder of this work.

If (2.2) is known, the inverse configuration $\boldsymbol{T}_i^j$ describing $\mathcal{T}^i$ with respect to $\mathcal{T}^j$ can be derived. First, note that the orthonormal property of the rotation matrix allows for $\boldsymbol{R}_i^j = \left(\boldsymbol{R}_j^i\right)^{\mathrm{T}}$. Then, the inverse configuration simply computes as

$$\boldsymbol{T}_i^j = \left(\boldsymbol{T}_j^i\right)^{-1} = \begin{bmatrix} \boldsymbol{R}_i^j & -\boldsymbol{R}_i^j \boldsymbol{p}_j^i \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix}, \tag{2.3}$$

refer to [88] for details.

The configuration matrices (2.2)–(2.3) allow to change the frame of reference for a position vector $\boldsymbol{p}^i$ in $\mathcal{T}^i$ with a single matrix multiplication to $\boldsymbol{p}^j$ in $\mathcal{T}^j$. For this purpose, the homogeneous representation of the position vector needs to be introduced as

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}. \tag{2.4}$$

With the help of (2.4) the change of the reference frame for a position vector is

$$\tilde{\boldsymbol{p}}^i = \boldsymbol{T}_j^i \tilde{\boldsymbol{p}}^j \quad \text{and} \tag{2.5}$$

$$\tilde{\boldsymbol{p}}^j = \boldsymbol{T}_i^j \tilde{\boldsymbol{p}}^i = \left(\boldsymbol{T}_j^i\right)^{-1} \tilde{\boldsymbol{p}}^i. \tag{2.6}$$

### 2.3.2 Kinematics of Free Ballistic Flight

Perceiving, estimating and describing the free flight of arbitrary objects is a challenging topic because occurring dynamics can range from chaotic behavior to quasi-linear motions. For example, a partially filled and rotating bottle impedes an analytical prediction of flight trajectories as the liquid state inside can barely be sensed or modeled. In contrast, a rigid steel object in free flight may only be significantly influenced by the constant gravitation while aerodynamic drag can be neglected.

In this thesis, the ballistic object flight of rigid objects is considered, allowing the research to focus on task and robot motion planning for robotic catching. Therefore, the mass-surface ratio is assumed to be sufficiently large, such that aerodynamic drag can be neglected. A filled aluminum ball, cuboid and flask-shaped object meet this assumption and are used as example objects in this research. Hence, free object flight is considered a linear dynamical system describing the motion of the object frame $\mathcal{T}^O$ with respect to the world frame $\mathcal{T}^W$ during free flight. Consider a world frame $\mathcal{T}^{\mathrm{W}}$ defined by orthogonal unit vectors $\hat{u}$, $\hat{v}$, and $\hat{w}$, as depicted in Figure 2.3. The first two unit vectors $\hat{u}$ and $\hat{v}$ are parallel to the floor. The third unit vector $\hat{w}$ is normal to the floor and thus aligned parallel with the direction of gravity $g$. An object flying in $\mathcal{T}^{\mathrm{W}}$

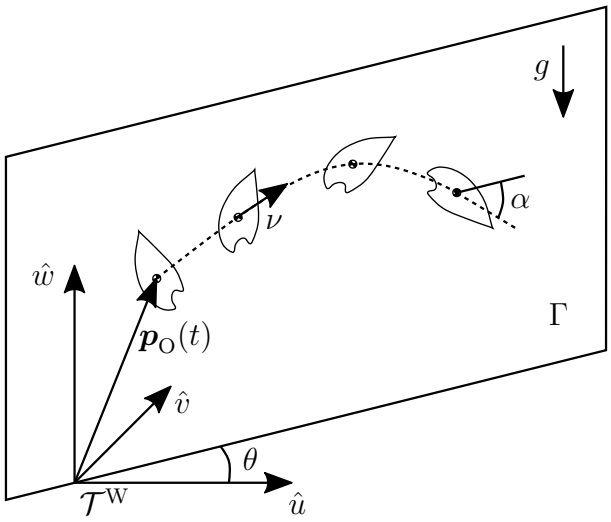


**Figure 2.3:** Free object flight, where the object's center of mass follows a ballistic trajectory in the flight plane Γ.

has its center of mass and therefore the origin of $\mathcal{T}^{\mathrm{O}}$ at

$$\boldsymbol{p}_{\mathrm{O}}(t) := \begin{bmatrix} u(t) \\ v(t) \\ w(t) \end{bmatrix} \tag{2.7}$$

with an orientation $\boldsymbol{R}_{\mathrm{O}}(t)$. Free flight translation under the assumption of negligible aerodynamic drag is therefore described by the kinematic differential equation

$$\ddot{u} = 0, \quad \ddot{v} = 0, \quad \ddot{w} = -g \tag{2.8}$$

with the initial positions, typically specified at the release or measurement instant $t = t_{\mathrm{r}}$, at

$$u(t_{\mathrm{r}}) = u_{\mathrm{r}}, \quad v(t_{\mathrm{r}}) = v_{\mathrm{r}}, \quad w(t_{\mathrm{r}}) = w_{\mathrm{r}}, \tag{2.9}$$

and initial velocities

$$\dot{u}(t_{\mathrm{r}}) = \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \cos(\theta), \tag{2.10a}$$
$$\dot{v}(t_{\mathrm{r}}) = \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \sin(\theta), \tag{2.10b}$$
$$\dot{w}(t_{\mathrm{r}}) = \nu_{\mathrm{r}} \sin(\alpha_{\mathrm{r}}), \tag{2.10c}$$

whereas $\nu_{\mathrm{O}}(t_{\mathrm{r}}) = \nu_{\mathrm{r}}$ is the initial object velocity in flight direction (alongside the flight path), the initial flight path angle with respect to the floor is denoted $\alpha(t_{\mathrm{r}}) = \alpha_{\mathrm{r}}$, and $\theta$ specifies the orientation of the flight plane Γ around $\hat{w}$.

Solving (2.8) with the initial conditions (2.9)-(2.10) by integration results in

$$\boldsymbol{p}_{\mathrm{O}}(t) := \begin{bmatrix} u(t) \\ v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} u_{\mathrm{r}} + (t - t_{\mathrm{r}})\nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \cos(\theta) \\ v_{\mathrm{r}} + (t - t_{\mathrm{r}})\nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \sin(\theta) \\ w_{\mathrm{r}} + (t - t_{\mathrm{r}})\nu_{\mathrm{r}} \sin(\alpha_{\mathrm{r}}) - \frac{1}{2} g (t - t_{\mathrm{r}})^2 \end{bmatrix}. \tag{2.11}$$

The velocity of the object's center of mass is therefore

$$\dot{\boldsymbol{p}}_{\mathrm{O}}(t) := \begin{bmatrix} \dot{u}(t) \\ \dot{v}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \cos(\theta) \\ \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \sin(\theta) \\ \nu_{\mathrm{r}} \sin(\alpha_{\mathrm{r}}) - g(t - t_{\mathrm{r}}) \end{bmatrix}, \tag{2.12}$$

which allows to compute the object velocity in flight direction by

$$\nu_{\mathrm{O}}(t) = \sqrt{\dot{\boldsymbol{p}}_{\mathrm{O}}^{\mathrm{T}}(t) \dot{\boldsymbol{p}}_{\mathrm{O}}(t)} \tag{2.13}$$

and the flight path angle by

$$\alpha(t) = \tan^{-1}\left( \frac{\dot{w}(t)}{\sqrt{\dot{u}^2(t) + \dot{v}^2(t)}} \right). \tag{2.14}$$

The object rotation $\boldsymbol{\omega} := \begin{bmatrix} \omega_u & \omega_v & \omega_w \end{bmatrix}^{\mathrm{T}}$ during free flight is constant as no accelerating forces are encountered. Hence, the orientation change is

$$\dot{\boldsymbol{R}}_{\mathrm{O}}(t) = \boldsymbol{R}_{\mathrm{O}}(t) \boldsymbol{S}(\boldsymbol{\omega}), \text{ where } \boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_w & \omega_v \\ \omega_w & 0 & -\omega_u \\ -\omega_v & \omega_u & 0 \end{bmatrix}, \tag{2.15}$$

which can be solved, for example, by numeric integration given an initial $\boldsymbol{R}_{\mathrm{O}}(t_{\mathrm{r}})$.

During free flight, a rotation $\theta$ exists such that the velocity in $\hat{v}$-direction becomes zero. This research assumes henceforth without loss of generality that

$$\theta = 0 \tag{2.16}$$

and $v = \dot{v} = 0$ hold at all times allowing planar treatment in the plane of flight $\Gamma$. The respective center of mass position of the planar system is denoted

$$\bar{\boldsymbol{p}}_{\mathrm{O}}(t) := \begin{bmatrix} u(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} u_{\mathrm{r}} + (t - t_{\mathrm{r}}) \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \\ w_{\mathrm{r}} + (t - t_{\mathrm{r}}) \nu_{\mathrm{r}} \sin(\alpha_{\mathrm{r}}) - \frac{1}{2} g (t - t_{\mathrm{r}})^2 \end{bmatrix} \tag{2.17}$$

and object velocities alongside the ballistic flight path simplify to

$$\dot{\bar{\boldsymbol{p}}}_{\mathrm{O}}(t) := \begin{bmatrix} \dot{u}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} \nu_{\mathrm{r}} \cos(\alpha_{\mathrm{r}}) \\ \nu_{\mathrm{r}} \sin(\alpha_{\mathrm{r}}) - g(t - t_{\mathrm{r}}) \end{bmatrix}. \tag{2.18}$$

**Remark 2.1.** *Assuming $\theta = 0$ and $v = \dot{v} = 0$ during free flight is mild because no acceleration due to gravity occurs in $\hat{v}$-direction. In the presence of collisions, assuming $\dot{v} = 0$ is only mild for spherical objects. Due to the poor predictability of polygonal collisions, instantaneous changes of $\dot{v}$ may occur. Motion in this direction, however, may be easily countered with a box-like end-effector design and the explicit consideration of uncertain collisions.*

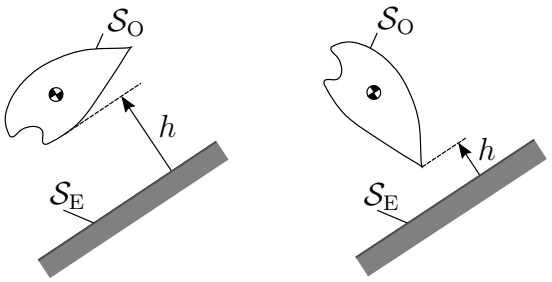


**Figure 2.4:** Distance measure $h$ between object surface $\mathcal{S}_\mathrm{O}$ and end-effector surface $\mathcal{S}_\mathrm{E}$. Rotations of the object lead to bounded uncertainty in $h$ for identical distances between the object center of mass and $\mathcal{S}_\mathrm{E}$.

### 2.3.3 General Catching Problem

Consider a generalized object with surface $\mathcal{S}_\mathrm{O}$ and a robot catching end-effector with desired catching surface $\mathcal{S}_\mathrm{E}$ (Figure 2.4) on the ballistic path introduced with Section 2.3.2. The catching surface $\mathcal{S}_\mathrm{E}$ can be of various types, e.g., a simple plate, the palm of a gripper, or the bottom of a box, as discussed later in this thesis. The distance between $\mathcal{S}_\mathrm{O}$ and $\mathcal{S}_\mathrm{E}$ is then defined as

$$h := \min_{\boldsymbol{p}_{\mathcal{S}_\mathrm{O}} \in \mathcal{S}_\mathrm{O}, \boldsymbol{p}_{\mathcal{S}_\mathrm{E}} \in \mathcal{S}_\mathrm{E}} \left\| \boldsymbol{p}_{\mathcal{S}_\mathrm{O}} - \boldsymbol{p}_{\mathcal{S}_\mathrm{E}} \right\|, \tag{2.19}$$

where $\boldsymbol{p}_{\mathcal{S}_\mathrm{O}}$ and $\boldsymbol{p}_{\mathcal{S}_\mathrm{E}}$ are points on the object and end-effector surface, respectively. Initially, surfaces $\mathcal{S}_\mathrm{O}$ and $\mathcal{S}_\mathrm{E}$ are disjoint such that the object is above the end-effector: $h > 0$. The time derivatives of $h$ are the relative velocity

$$\nu := \tfrac{\mathrm{d}}{\mathrm{d}t} h \tag{2.20}$$

and the relative acceleration

$$\gamma := -\tfrac{\mathrm{d}^2}{\mathrm{d}t^2} h. \tag{2.21}$$

Hence, the goal of catching an arbitrarily shaped, fast flying rigid object corresponds to

$$h(t \to \infty) = 0 \quad \text{and} \quad \nu(t \to \infty) = 0, \tag{2.22}$$

where $\gamma > 0$ must hold.[2]

In contrast to the distance definition for nonprehensile manipulation in [56], the approach (2.19) always selects the closest point on $\mathcal{S}_\mathrm{O}$ instead of measuring the distance to the object's center of mass. Object rotation, thus, causes considerable uncertainties to the distance measurement and its derivatives. These uncertainties, however,

[2] In practice, one may replace the asymptotic formulation in (2.22) with a finite time goal depending on the individual setup. Section 3.4.3 discusses this issue by means of *Zeno behavior* resulting in (3.37).

are formally considered as defined in Section 2.3.5 and analyzed by a simulation in Section 3.5.2. The clear advantage of formulation (2.19) is that the occurrence of collisions simplifies to the condition $h = 0$ and $\nu < 0$, which is the key for closing the gap between spatial catching and the one-dimensional hybrid bouncing ball detailed in Chapter 3. Most importantly, this approach shifts object orientation and rotation from the formal relative description to the scenario uncertainties ultimately leading to intuitive solutions that are easy to implement.

Another major difference in this thesis to existing catching work is the goal definition (2.22). Previous work on dynamic catching has typically targeted an exact match of an object and end-effector velocity at an interception point [52, 78]. Such a goal cannot be achieved formally in practice as some uncertainty always remains. Achieving (2.22), however, is more realistic as this condition is not immediately violated in the presence of uncertainties and may encompass the inevitable sequence of collisions.

### 2.3.4 Collision Modeling

A collision is an omnipresent phenomenon that appears when two bodies contact ($h = 0$ in (2.19)). In a robotic context, popular examples are non-negligible ground impacts of walking robots or batting and catching robots in the field of dynamic manipulation. Prior to such contact, the contact points on the surfaces of each body converge with a non-zero relative velocity, which persists to some extent at the instant of contact. Hence, reacting forces and torques inevitably occur, which modify the motion of the affected bodies, also referred to as a *collision*. During such collisions, the contacting bodies are subject to elastic or plastic deformation, or both, accompanied by dissipation of energy [36]. The forces are comparably high as the contact duration is typically very brief, i.e. in the order of a few milliseconds. This short duration together with the large force magnitudes constitute a modeling problem highly sensitive to parameter uncertainty. Bodies (objects) that are not spherical or experience multi-contact collisions, as in the first strike of a game of pool, have particularly challenged, and still test, many researchers.

In this section, common classification criteria and collision models are reviewed and briefly discussed in the context of object catching. The basis of many works up until today has constituted Routh's work [73], which proposes a graphical, impulse-based analysis. An updated and extensive introduction to the field of non-smooth mechanics is included in Brogliato's book [15]. Coutinho [23] and Brach [13] provide alternatives to Brogliato's book for further reading and with more focus on impacts. The literature review conducted by Gilardi and Sharf [33] categorizes and reviews the topic more concisely, whereas Stewart [91] provides a rather extensive mathematical view combined with illustrative examples.

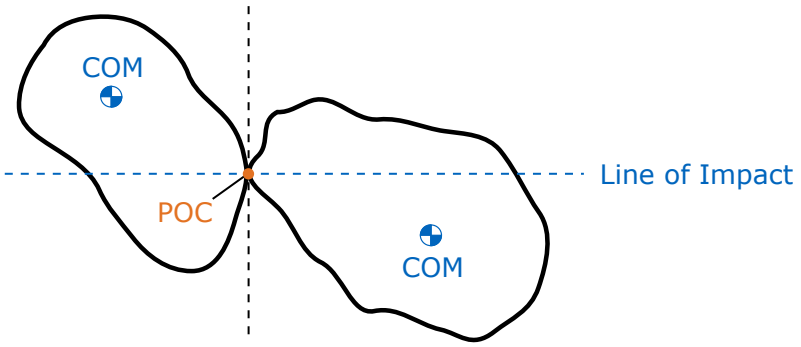


**Figure 2.5:** Impact classification and modeling depends on the center of mass (COM) locations with respect to the line of impact. The line of impact represents the normal direction in the point of contact (POC).

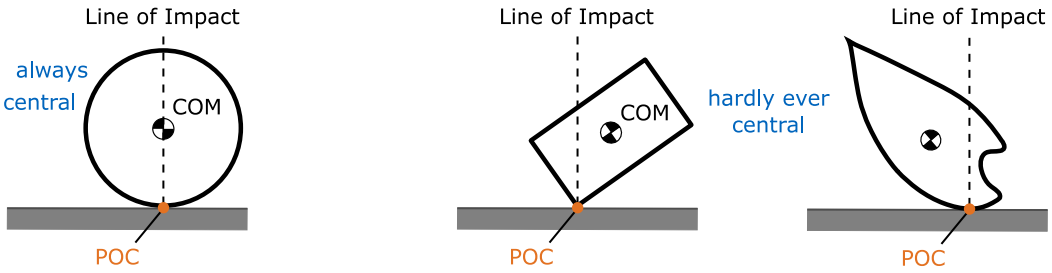


**Figure 2.6:** Impacts between spherical objects with equal mass distribution and a plate are always central. Impacts between cuboids or other objects with unequal mass distribution and a plate are rarely central.

### Impact Classification

Consider two bodies colliding at one point, as depicted in Figure 2.5. In this case, there exists an infinitesimal contact surface defining a tangential contact plane[3] with a respective normal, i.e.,the *line of impact*. Velocities at the point of contact, before and after impact, can be separated into a normal velocity parallel to the line of impact and one (2D) or two (3D) tangential velocities perpendicular to the line of impact.

If the center of mass for both objects are positioned on the line of impact, a *central* or *collinear* impact applies. In this case, the normal velocity component is decoupled and, therefore, does not influence tangential velocities or rotation, and vice versa. Colliding rigid spherical objects with equal mass distribution always constitute a central impact. The same applies for the collision of such objects with a plate, as in the left example of Figure 2.6, that can be assumed to have a comparably infinite mass, for example, a ball bouncing on a floor or table. In both of these special cases for spherical objects, the impact modeling problem is no longer sensitive to the relative center of mass position.

[3]If the contact point of one of the bodies is an ideal vertex, the contact plane may be defined by the second object [96]. The rare problem of double-vertex impacts is neglected.

Hence, impact modeling and the prediction of an impact outcome are simpler and much more reliable.

If the center of mass for one or both objects are not positioned on the line of impact, an *eccentric* impact applies. In this case, the normal velocity may induce or reduce tangential velocity and rotation, or vice versa. Here, a small change in the relative center of mass position may cause large variations in the post-impact state of both colliding objects. For most of the non-spherical objects, as in the two right-hand examples in Figure 2.6, eccentric impacts are the rule rather than the exception. For example, throwing a dice repeatedly with the same result is almost impossible, whereas the same task is rather simple using a partially elastic ball, such as a basketball [8].

Further classification regards the velocities at the impact point. A *direct* impact applies, if the initial contact velocity of both bodies is aligned with the line of impact, i.e., there are no tangential velocity components. If one or both velocities are not aligned with the line of impact, an *oblique* impact applies. Sometimes, *tangential* impacts [96] are defined in addition to complementing the previous two definitions with the mostly theoretical case of no velocity components in the normal direction.

### Energy Loss

The goal of impact modeling is to predict the post-impact states of colliding bodies based on pre-impact states and other parameters. The particular selection of these other parameters provides a distinction between the large number of proposed techniques, including the common examples of the geometry of bodies, material properties, contact duration and friction [36].

Two phases in the deformation process during one collision event are distinguished as *compression* and *restitution*. The compression phase begins when the bodies contact and persists until the maximum deformation is reached. The restitution phase starts at the moment of maximum deformation and continues until the bodies separate. Energy is lost depending on the aforementioned parameters the define a collision system, for example, in non-recoverable deformation, sound or heat [8]. Typically, a coefficient of restitution $0 \leq e \leq 1$ is defined to describe this loss of energy. The following terms have been established to describe the range of possible energy loss and deformation:

- *Perfectly elastic* ($e = 1$), where no energy is lost.
- *Perfectly plastic* ($e = 0$), where all energy is lost and deformation is permanent.
- *Partially elastic* ($0 < e < 1$), with partial energy loss, but without significant permanent deformation.
- *Partially plastic* ($0 < e < 1$), with partial energy loss and permanent deformation.

As such, perfect elasticity is of a rather theoretical nature, whereas partial elasticity or plasticity constitute the most common cases. Nevertheless, nearly perfect plasticity

might also be achieved in practical scenarios, for example, by covering a body with slow recovery foam [57].

### Impact Models

Modeling impacts is a complex challenge characterized by large forces acting over a very brief time duration. Also, other physical phenomena occur, for example, vibrations, inner-body strain, locally varying persistent and non-persistent deformations, and several types of friction. All these phenomena can cause a considerable amount of energy loss and a change of the post-impact state in the system of colliding bodies. Most work aiming for precise impact modeling, therefore, is limited to rigid bodies because otherwise, expensive simulations of inner-body processes, e.g., using finite element methods [6, 7], are necessary. Nevertheless, previous research has demonstrated that models based on the rigid body assumption are sufficiently precise to describe a significant amount of real systems [96].

During the very brief impact duration, a normal force $F$ acts along the line of impact. Due to its nearly instantaneous nature, this contact force is typically associated with an infinitesimally short time integral (Dirac's delta function)

$$P = \lim_{\Delta t \to 0} \int F(t) \mathrm{d}t \quad , \tag{2.23}$$

termed as an *impulse*. By definition, this integral is finite.

The total impulse $P$ during an impact can be separated into a compression impulse $P_\mathrm{c}$ and restitution impulse $P_\mathrm{r}$. The loss of energy during an impact may then be represented according to *Poisson's law of restitution* with the kinetic coefficient of restitution defined by

$$e_\mathrm{P} = \frac{P_\mathrm{r}}{P_\mathrm{c}}. \tag{2.24}$$

Considering that $P = P_\mathrm{c} + P_\mathrm{r}$, this approach asserts that impulse gained during restitution does not exceed the impulse generated during compression. This law is widely applied to frictional impacts in combination with Coulomb's law of friction using Routh's graphical impulse analysis [73]. Jia extended the impulse analysis to three-dimensions using virtual springs and a system of differential equations including impulse and energy terms [43].

A second, common model follows *Newton's law of restitution*, which defines a kinematic coefficient of restitution

$$e_\mathrm{N} = \frac{V^+}{V^-}, \tag{2.25}$$

where $V^-$ and $V^+$ are the normal velocities before and after a collision at the contact point. These velocities are defined along the line of impact. Although a simple description of the complex impact process, the model (2.25) is often sufficiently accurate, especially in cases of direct and central impacts, e.g., a vertically bouncing ball. In the

presence of complex bodies, however, situations may be created in which Newton's law violates conservation of energy principles [96].

This research applies Newton's law (2.25) for its simplicity and to build on its wide spread use in control theory. Knowing that this model is imprecise when describing impacts of arbitrarily shaped objects, the coefficient of restitution is considered uncertain, which is also reflected in the use of

$$\rho := e_{\mathrm{N}} \tag{2.26}$$

with $\rho \in [0, 1)$. The rare situations in which energy principles are violated ($\rho \geq 1$), are neglected hereafter.

### 2.3.5 Generalized Uncertainty Formulation

In a catching scenario, various uncertainties occur and are difficult to measure or predict. Nonetheless, boundedness can be assumed, and the following three types are distinguished to account for these uncertainties:

U1) Uncertainty in the initial state $h(0)$ and $\nu(0)$ defined with (2.19) and (2.20) due to, for example, imprecise object state estimation including orientation and rotation gained from real-time vision or based on the repeatable, open-loop robot throws as in this research.

U2) Time-varying uncertainties in $\gamma(t)$ defined with (2.21) due to, for example, object rotation, neglected aerodynamic drag, imprecise robot motions, and object motions deviating from the ballistic path. Errors in the estimation of $h$ and $\nu$ for $t > 0$ as a consequence of U1 lead to translational object motion in parallel to $\mathcal{S}_{\mathrm{E}}$ (perpendicular to $h$), which in turn affects the relative acceleration.

U3) Collisions in catching are inevitable due to U1 and U2. The outcome of such collisions is unpredictable and thus, the kinematic restitution $\rho$ defined with (2.25)-(2.26), which acts perpendicular to $\mathcal{S}_{\mathrm{E}}$, must also be considered as uncertain.

From the problem formulation defined in Section 2.3.3 and uncertainties U1-U3, it follows that catching an arbitrarily shaped, fast-flying rigid object is similar to stabilizing a one-dimensional bouncing ball with considerable uncertainties. Thus, this research concentrates on generating robustness against U1-U3 generally, instead of calculating scenario-specific bounds for all uncertainties.

**Remark 2.2.** *The acquisition and forward projection of an object orientation* $\boldsymbol{R}_{\mathrm{O}}(t)$ *and rotation* $\dot{\boldsymbol{R}}_{\mathrm{O}}(t)$ *defined with* (2.15)*, is an expensive perception challenge. Therefore, given uncertainties U1-U3 and the distance definition* (2.19)*, the object orientation* $\boldsymbol{R}_{\mathrm{O}}(t)$ *is neglected throughout all task planning stages hereafter. Instead, U1 and U2 account for the range of possible orientations and rotations. The distance variation due to unknown orientations is naturally bounded and rotations are assumed to be bounded.*

## 2.4 Robot Motion Planning Problem

This section introduces fundamental models describing the robot motion. Section 2.4.1 presents geometric transformations for describing a manipulator configuration. Section 2.4.2 briefly reviews manipulator dynamics and constraints that must be met to guarantee accurate robot motions. Section 2.4.3 reviews the concept of kinematic redundancy, which provides the basis for Chapter 4.

### 2.4.1 Robot Kinematics

A manipulator is an articulated robotic structure consisting of a series of connected rigid bodies. The connections are called *joints*. These joints can either be directly controllable with a motor (*actuated*, *active*) or can be *unactuated* (*passive*). Passive joints are influenced indirectly through the motion of active joints, external forces, and the dynamics of connected rigid bodies. Together, active and passive joints constitute the kinematic chain of a manipulator. An extensive discussion of the various types of kinematic chains is provided in in [88].

Consider a manipulator with $m$ joints, also referred to as an $m$ degrees of freedom (DOF) robot. The state of this $m$-DOF robot is represented by its joint displacements $\boldsymbol{q}_\mathrm{m} \in \mathbb{R}^m$, which can be angles in case of revolute joints or distances in case of linear joints. At one end, the manipulator is attached to the environment, where the world frame $\mathcal{T}^\mathrm{W}$ may also be defined. At the other end, an end-effector (tool, gripper) is attached to the manipulator with its respective frame $\mathcal{T}^\mathrm{E}$. The relation between the two frames, depending on a particular configuration $\boldsymbol{q}_\mathrm{m}$, may then be expressed with a transformation

$$\boldsymbol{T}_\mathrm{E}^\mathrm{W}(\boldsymbol{q}_\mathrm{m}) = \boldsymbol{T}_1^\mathrm{W}(q_1)\boldsymbol{T}_2^1(q_2)\dots\boldsymbol{T}_m^{m-1}(q_m)\boldsymbol{T}_\mathrm{E}^m, \tag{2.27}$$

as introduced in (2.2). The final multiplier $\boldsymbol{T}_\mathrm{E}^m$ defines the end-effector frame $\mathcal{T}^\mathrm{E}$ with respect to the end of the kinematic chain and is often included in the final transformation $\boldsymbol{T}_m^{m-1}(q_m)$. If the end-effector device is exchanged, the pose $\boldsymbol{T}_\mathrm{E}^m$ might also change. Moreover, it should be noted that the computation of

$$\boldsymbol{T}_m^{m-1}(q_m) = \begin{bmatrix} \boldsymbol{R}_m^{m-1}(q_m) & \boldsymbol{p}_m^{m-1}(q_m) \\ \boldsymbol{0}^\mathrm{T} & 1 \end{bmatrix} \tag{2.28}$$

is not unique and conventions for its computation exist, for example, the Denavit-Hartenberg Convention [88].

Combining the kinematic transformation (2.27) with (2.2)-(2.18) enables the computation of the relative pose of an object with regard to the end-effector

$$\boldsymbol{T}_\mathrm{O}^\mathrm{E}(\boldsymbol{q}_\mathrm{m}, t) = \boldsymbol{T}_\mathrm{W}^\mathrm{E}(\boldsymbol{q}_\mathrm{m}(t))\boldsymbol{T}_\mathrm{O}^\mathrm{W}(t) \tag{2.29}$$

in robotic scenarios. As a result, the distance $h$ defined in (2.19) and its derivatives can be simulated.

### 2.4.2 Robot Dynamics and Joint Constraints

The kinematic definitions in Section 2.4.1 only describe the end-effector configuration concerning the world frame in a geometric way. The manipulator's mass and inertia are disregarded. A body of work on robotic catching settles for this limitation of mass negligence to simplify robot motion planning [10, 17, 78]. Experiments such as in [78], however, have demonstrated that neglecting any mass-related properties might be overly restrictive in the case of dynamic catching. Therefore, this research considers the joint space dynamic model

$$\boldsymbol{M}(\boldsymbol{q}_\mathrm{m})\ddot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{C}(\boldsymbol{q}_\mathrm{m}, \dot{\boldsymbol{q}}_\mathrm{m})\dot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{G}(\boldsymbol{q}_\mathrm{m}) = \boldsymbol{\tau} \ , \tag{2.30}$$

with the inertia matrix $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, Coriolis matrix $\boldsymbol{C} \in \mathbb{R}^{m\times m}$, gravitational vector $\boldsymbol{G} \in \mathbb{R}^m$ and the input torque $\boldsymbol{\tau} \in \mathbb{R}^m$. An extensive derivation of the joint space dynamic model is detailed in [88]. External forces acting on the robotic manipulation system and frictional energy loss in the joints are neglected during robot motion planning in this thesis. For collision modeling, as described in Section 2.3.4, negligible object mass equals a theoretically infinite mass of the end-effector.

Having planned a task motion for an end-effector such that the task goal is robustly achieved within pre-defined limits, it is crucial to repeatedly accurate perform this end-effector motion with a manipulator. Failure in this regard renders void any guarantees made at the task level. The following constraints must be maintained to achieve repeatedly accurate manipulator motions:

- **Joint range limits** originate in the kinematic structure of a robot manipulator. For commercial manipulators, these limits are provided by the manufacturer. Their purpose includes, but is not limited to, avoiding that subsequent parts in the kinematic chain hit the robot itself or preventing cables from twisting. For all joints, a lower and upper bound for their range may be defined aggregated in $\underline{\boldsymbol{Q}}$ and $\overline{\boldsymbol{Q}}$, respectively:
$$\underline{\boldsymbol{Q}} \leq \boldsymbol{q}_\mathrm{m} \leq \overline{\boldsymbol{Q}}. \tag{2.31}$$
- **Joint velocity limits** depend on the actuation hardware. Manufacturers of classical actuators, for example, linear motors or motors connected to gears, specify these upper (positive) and lower (negative) limits hereafter aggregated in $\underline{\boldsymbol{V}}$ and $\overline{\boldsymbol{V}}$, respectively, such that
$$\underline{\boldsymbol{V}} \leq \dot{\boldsymbol{q}}_\mathrm{m} \leq \overline{\boldsymbol{V}} \tag{2.32}$$
must hold. Violating these velocity limits during robot motion planning usually causes unpredictable deviations of the end-effector from its desired trajectory.
- **Joint torque limits** also depend on the actuation hardware and must not be exceeded so as to avoid task trajectory degeneration or actuator damage. Upper and lower torque limits are aggregated in $\underline{\boldsymbol{T}}$ and $\overline{\boldsymbol{T}}$, respectively, such that
$$\underline{\boldsymbol{T}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{T}} \tag{2.33}$$

must hold. Moreover, the limitation of joint torque indirectly limits joint acceleration through (2.30).

- **Motor power limits** can also become relevant during fast manipulation tasks such as dynamic catching. As dynamic catching requires quick joint motions close to the limits (2.32) while the end-effectors follows the free flight trajectory, motor power limits are often more restrictive than the peak torque limits (2.33). The variable $\overline{\boldsymbol{P}}$ represents these power limits in

$$\dot{\boldsymbol{q}}_{\mathrm{m}}\boldsymbol{\tau} \leq \overline{\boldsymbol{P}}. \tag{2.34}$$

  Similar to the previous two limit types, violating the power limits during robot motion planning usually causes unpredictable deviations of the end-effector from its desired trajectory. Due to the use of lossy gears and a power consideration at a load-side mechanical level, large frictional effects support deceleration at high velocities on the motor side of the gears. The negative peak torque limit is, therefore, in practice more restrictive in most relevant situations than a negative power limit.[4]

### 2.4.3 Kinematic Redundancy

Consider a task given by $\boldsymbol{x}_{\mathrm{n}} \in \mathbb{R}^n$, where $n$ denotes the number of variables necessary to describe the task, for instance, an end-effector motion. Here, the task space related to $n$ can be of a lower dimension than the operational space. For example, a task that only defines a planar end-effector position in 3D space has $(n = 2)$ as one translation and all orientations are not specified. Thus, the concept of kinematic redundancy describes the relative situation between such an $n$-dimensional task and an $m$-DOF manipulator meant to perform this task. In case of $(n = m)$, no redundancy exists. In case of $(n < m)$, the kinematic structure provides redundancy regarding the particular task $\boldsymbol{x}_{\mathrm{n}}$.

Now, let $\boldsymbol{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ describe the map from joint space to task space. Then, given the previously introduced manipulator configuration $\boldsymbol{q}_{\mathrm{m}}$, the $(n \times m)$ Jacobian is defined as $\boldsymbol{J}(\boldsymbol{q}_{\mathrm{m}}) = \partial \boldsymbol{f}(\boldsymbol{q}_{\mathrm{m}})/\partial \boldsymbol{q}_{\mathrm{m}}$. The differential kinematic equation is

$$\dot{\boldsymbol{x}}_{\mathrm{n}} = \boldsymbol{J}(\boldsymbol{q}_{\mathrm{m}})\dot{\boldsymbol{q}}_{\mathrm{m}}. \tag{2.35}$$

Hence, this Jacobian enables calculation of the resulting end-effector velocity in task space given a manipulator joint configuration and velocity.

An important property of (2.35) in case of redundancy is the existence of a *null space* set $\mathcal{N}(\boldsymbol{J})$ of possible joint velocities $\dot{\boldsymbol{q}}_{\mathrm{m}}$ that result in the same task space velocity $\dot{\boldsymbol{x}}_{\mathrm{n}}$,

[4] Most robot motions generated in this research directly profit from these frictional effects during deceleration. Adding the overly restrictive negative power limit without including adequate friction in the dynamic model would result in empty solution spaces for the robot motion planners presented in this research.

given a particular Jacobian $\boldsymbol{J}$. The dimension of this null space set for non-singular Jacobians is $\dim(\mathcal{N}(\boldsymbol{J})) = m - n$. As the present research only considers methods that aim for feasible motions internally, the critical issue of singular configurations does not need to be covered here. For extensive analysis and discussion of singularity and methods solving related problems refer to [88].

As a consequence of the introduced null space, an $(m \times m)$ projection matrix $\boldsymbol{P}$ exists such that $\boldsymbol{J}\boldsymbol{P}\dot{\boldsymbol{q}}_{\mathcal{N}} = \boldsymbol{0}$ holds for arbitrary $\dot{\boldsymbol{q}}_{\mathcal{N}} \in \mathbb{R}^m$. The kinematic equation (2.35) can, therefore, be modified to

$$\dot{\boldsymbol{x}}_{\mathrm{n}} = \boldsymbol{J}(\boldsymbol{q}_{\mathrm{m}})(\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{P}\dot{\boldsymbol{q}}_{\mathcal{N}}) \tag{2.36}$$

without changing the resulting task space velocities.

The goal of robot motion planning, however, is to solve the inverse problem of (2.35) or (2.36) to identify a joint space motion

$$\dot{\boldsymbol{q}}_{\mathrm{m}} = \boldsymbol{J}^{-1}(\boldsymbol{q}_{\mathrm{m}})\dot{\boldsymbol{x}}_{\mathrm{n}} \tag{2.37}$$

and make use of the redundancy to fulfill additional sub-tasks.

## 2.5 Summary

Catching flying objects with a robotic manipulator is a testbed frequently considered when evaluating high-performance perception and motion planning solutions. Static catching and dynamic catching can be distinguished, whereas hard impacts characterizes static approaches and dynamic approaches follow the object flight for some time to achieve softer contacts. The complexity of the catching challenge demands a clear structure and definition of the overall problem as given by the general catching problem formulation detailed in this chapter. Moreover, it is essential to identify accurately subproblems that are ideally decoupled. For some subproblems, adequate solutions may already exist, but for the others, extensions or new approaches are necessary, which constitute the focus for the remainder of this research.

Two conceptual dimensions are introduced in this chapter to efficiently discuss the interconnectivity of the robotic catching problem: two levels and four phases. The task planning level focuses on the relative system between the catching device (end-effector) and flying objects. The second conceptual dimension considers time and divides both levels into four phases: throwing, catching acceleration, flight trajectory tracking and deceleration.

At the task level, the inevitable occurrence of impacts in a catching process is explicitly considered, while the particular robot structure (manipulator) is disregarded. A novel problem definition enables explicit and efficient uncertainty treatment by using the shortest distance between the object's surface and end-effector's surface. The joint level planning focuses on the robot motion, which must provide a reliable performance

of the desired task plan. Reliable task performance might be achieved by guaranteeing that all crucial constraints of motors and gears are not violated. The formal concept of relative redundancy in a manipulator structure regarding the desired task provides the leverage point for the remainder of this research to realize flexible and reliable task motions.

# 3 Robust Hybrid Bouncing Ball in Ballistic Robotic Catching

This chapter addresses the gap between current academic catching robots and their future in industrial applications, specifically, reliable task execution. A novel parametrization is derived to reduce the three-dimensional catching problem from 3-D to 1-D on the ballistic flight path. Inversely, a real-time capable dynamical system formulation allows reconstruction of solutions from one dimension to three dimensions. Hence, the body of work on hybrid dynamical systems theory, in particular the one-dimensional bouncing ball problem, becomes available for robotic catching. Uniform Zeno asymptotic stability (UZAS) from the bouncing ball literature is adapted to enable the associated notion for an exact tight bound on the maximal Zeno time. Using explicit and finite maximal Zeno time, a special catching task problem for robotics is formulated to overcome weaknesses of the general catching goal with a potentially infinite time horizon. Moreover, an extension of the maximal Zeno time proposes a closed-form for desirable initial object states relative to the end-effector motion such that distance uncertainty compensation is maximized. Another extension, commencing from UZAS, proposes a bound on the maximal rebound height, which is potentially less conservative if generic box-like end-effectors are used as presented in Chapter 5. Finally, the uncertainty contribution of object shapes is discussed and evaluated based on a realistic simulation example.

## 3.1 Introduction and State of the Art

The fundamental problem of catching is to bring a fast flying object to rest by interaction with a robot end-effector. Previous work, however, has focused on perception and prediction of object flight [31, 46], well-timed interception [11] of ballistic flight or precise matching of the object state with a gripper [78]. Thus far, no ballistic catching approach has formally included the inevitable appearance of impacts, which are caused by uncertain knowledge of the object state. A preliminary analysis [85] has demonstrated via simulation and experiments that uncertain object states may lead to unfavorable relative motions if pure velocity matching is pursued. A reachability analysis also indicated that proper treatment of relative acceleration has the potential to perform graspless catching on ballistic trajectories. These results are in line with the findings of Schaal *et al.* [81] and more recently, Ronsse *et al.* [71] and Reist *et al.* [69] who revealed that negative acceleration has a focusing effect on manipulation with re-

bounds. In this chapter, the treatment of relative acceleration in robotic catching is formalized by a Lyapunov-based stability analysis of the fundamental hybrid system dynamics, which expresses the problem of convergence to a fixed point.

The problem of a ball bouncing on a table under a constant gravitational field is highly related to catching with rebounds. As a generic example for a process that is of a partially continuous (flow) and discrete (jump) nature, the bouncing ball is not only an illustrative example but has also motivated a body of work. A commonly revisited problem is stabilizing the ball on a periodic orbit in the example of robotic juggling [12, 15, 17, 69–72, 76, 79, 82, 94, 99]. One approach is to use measure differential equations [15] to model and analyze this problem. Most of the juggling approaches, however, have applied a Poincaré map approach [15, 17, 69, 71, 72, 99] to address the problem of periodic stability. These Poincaré map approaches have modeled the continuous phase with the parabolic solution in the world coordinates and then analyzed the implications of intermittent contacts with a , mostly periodically, moving table. Thereby, the work in [69, 71] confirmed Schaal's early observation in [82] of the significant influence of table acceleration on orbital stability behavior.

Today, progress in hybrid dynamical system theory has enabled a unified treatment of partially continuous and partially discrete problems. An extensive overview of the hybrid dynamical system framework used in the following and related stability for time-invariant hybrid systems is presented in [12, 34, 79] and includes applications of this framework on the juggling problem.

For the graspless robotic catching examined in this thesis, the fundamental underlying problem of *making the ball converge* to a fixed point on the table or end-effector [56] differs from juggling. The main difference is the occurrence of *Zeno behavior*, which is the unique ability of hybrid systems to exhibit an infinite number of discrete events during finite periods. The literature on Zeno behavior often focuses on conditions for its existence [19, 38, 101] for general, and even nonlinear, hybrid systems and how this existence correlates to asymptotic stability [1, 35, 63]. The generality of these approaches, however, hinders explicit calculations of Zeno limit points or finite Zeno time for particular initial conditions. The works [49, 64] constitute significant improvement towards an explicit consideration of Zeno behavior in real-world robotics. Following the work of Goebel *et al.* [35], they demonstrate general Zeno stability for a non-autonomous bouncing ball system with set-valued relative acceleration. Concentrating on the relative dynamics between the constraint and ball, the authors derive a necessary and sufficient condition for *uniform Zeno asymptotic stability* (UZAS) in [64] with a ratio between acceleration and the kinematic coefficient of restitution. The success guarantee and extensions presented in this chapter build on this ratio and the sophisticated Lyapunov function introduced for their proof. The contributions of this chapter are as follows:

(i) Dimensionality reduction closes the gap between ballistic catching and the related hybrid control theory, enabling the transfer of (future) progress in hybrid control theory to robotic catching.

(ii) As a consequence of contribution (i), work on uniform Zeno asymptotic stability (UZAS) and maximal Zeno time [64], both for the one-dimensional set-valued bouncing ball problem, is extended for robotic catching:
   (a) to proof robustness against an uncertain kinematic coefficient of restitution,
   (b) to derive a closed-form solution that maximizes the distance uncertainty compensation, which significantly increases catching robustness, and
   (c) to predict the maximal remaining rebound height of an uncertain hybrid bouncing ball.

(iii) The generality of the presented robustness, represented by the three types U1-U3, covers most of the practical uncertainty sources without the need to identify individual magnitudes.

(iv) The explicit consideration of inevitable Zeno behavior in realistic robotic catching enables reformulation of the general catching goal (2.22) with an infinite time horizon as a finite time horizon problem to suit realistic robotic scenarios.

(v) A simulation study reveals that a significant offset in the relative acceleration, which results from previously neglected deviations of the object from its ballistic path, is covered by the set-valued approach.

When continuous contact between an object and unilateral constraint, both with state uncertainty, is to be established, a sequence of collisions during finite time inevitably applies. This is the case because the individual uncertainties neither allow for precise distance measurement nor for precise relative velocity measurement or estimation, which ultimately causes at least small rebounds. Building on this fundamental observation, the methods and discussions presented in this chapter contrast previous work, which has typically aimed for ideal (i.e., without any rebounds) transitions into continuous contact by velocity matching. As such, these previous works lack formulating conditions that must be fulfilled to guarantee asymptotic stability (i.e., convergence) during this non-instantaneous transition. The hybrid dynamical system framework briefly reviewed in Section 3.2 is well suited to formulate the stability problem for a decoupled one-dimensional bouncing ball, including the notion of Zeno behavior.

Robotic catching is governed by two distinct types of problems: the relative motion between an object and end-effector on the one hand, and the kinematic motion of the robot on the other. The dynamics and constraints of both problems differ and make the overall solution highly complex. Therefore, Section 3.3 first presents a dimensionality reduction to decouple these two problems with an efficient interface. Based on

the assumption of a ballistic trajectory and a known object state, the object motion in the direction of the ballistic path (direction of flight) is also known. Combined with an initial end-effector state and acceleration on this path, one-dimensional bouncing ball dynamics in the flight direction can be mapped to the bouncing ball problem, which has been extensively researched by the hybrid systems and control community. Given an end-effector trajectory, the ballistic bouncing ball's stability, convergence and robustness can be analyzed. Alternatively, a one-dimensional end-effector trajectory can be designed to achieve certain stability or robustness goals. A dynamical system motion planner, with the end-effector acceleration as input, presented in subsequence, efficiently reconstructs the three-dimensional end-effector trajectory from the one-dimensional solution by once again using the ballistic assumption. This dimensionality reduction and reconstruction has been published in [84].

In particular, related work on uniform Zeno asymptotic stability for a set-valued bouncing ball and an exact tight bound on the maximal Zeno time [64] are proposed as a basis to formalize catching robustness (Section 3.4). The first of three extensions addresses uncertainty in the restitution behavior. In brief, it is derived that a trajectory of the unilateral constraint, which provides UZAS for a particular coefficient of restitution, is also UZAS for all coefficients smaller than this particular value. Moreover, the maximal Zeno time associated with trajectory and coefficient of restitution is also valid as an upper bound for all lower values of the coefficient of restitution. Hence, a single catching motion potentially leads to robust catching for a range of object materials. The second extension proposes a beneficial relative initial velocity based on task parameters such as coefficient of restitution, acceleration bounds and desired maximal Zeno time (maximal possible tracking duration: P3). The presented closed-form relative velocity maximizes the compensation of distance uncertainty. The third extension derives a bound on the maximal remaining rebound height for UZAS solutions. A numeric example in Section 3.4.7 illustrates possible interpretations of the robustness extensions and provides a basis for the experimental validation detailed in Chapter 5. UZAS in the context of robotic catching and its presented extensions have been published in [84].

The final part of this chapter is devoted to an evaluation of dynamics this research neglected in the process of dimensionality reduction. Uncertain knowledge of the object state not only reflects in imprecise distance estimation but also in the ballistic trajectory deterioration. In Section 3.5, this deterioration is considered to cause motions perpendicular to the expected flight trajectory. Errors in the end-effector orientation as in Chapter 5, rotating objects, or non-spherical object shapes can significantly influence such perpendicular motions at collisions. Considering the one-dimensional bouncing ball, these deviations cause an offset in the actual relative acceleration, which is quantified with a realistic physical simulation. Parts of the uncertainty evaluation have been published in [84].

## 3.2 One-Dimensional Hybrid Dynamics of the Bouncing Ball

In this section, the one-dimensional hybrid dynamics of a bouncing ball are derived starting with a general definition of the selected hybrid system framework (Section 3.2.1). Moreover, the relation to other frameworks is discussed briefly. Section 3.2.2 provides a short introduction of the so-called *Zeno behavior*, which is a unique property of hybrid dynamical systems and crucial when contacts are to be robustly established. Section 3.2.3 formulates the hybrid bouncing ball dynamics such that the acceleration of a unilateral constraint can be considered as input to this system.

### 3.2.1 Hybrid System Formulation

The hybrid system formulation presented in the following originates from the framework [34], which contains a recommendable introduction to the topic of hybrid dynamical systems. This formulation is widely used in the current hybrid systems literature and is especially common when impulsive systems are regarded, as is the case in this thesis. Findings from [34] also demonstrate that other frameworks for hybrid systems [14, 16], still in use for other problem settings such as hybrid automata [65, 87], switched systems [50], sampled data [28] and networked control systems [95], can be cast as a hybrid system of the following form with hybrid dynamics defined thereafter.

**Definition 3.1.** *A hybrid system is described by the 4-tuple*

$$\mathcal{H} := (\mathcal{C}, \mathcal{F}, \mathcal{D}, \mathcal{G}), \tag{3.1}$$

*where $\mathcal{C}$ is a subset of the Euclidean space describing the continuous domain, $\mathcal{F}$ is a set-valued mapping describing the continuous dynamics, $\mathcal{D}$ is a subset of the Euclidean space describing the discrete domain, and $\mathcal{G}$ is a set-valued mapping describing the discrete dynamics.*

Hybrid systems $\mathcal{H}$ allow for a combined treatment of physical systems governed by both continuous and discrete dynamics. Hybrid phenomena can have various sources in Definition 3.1, such as the multi-valuedness of the mappings $\mathcal{F}$ and $\mathcal{G}$. However, hybrid phenomena usually come from the geometry of sets $\mathcal{C}$ and $\mathcal{D}$ [34], as it is the case in this thesis.

**Definition 3.2.** *A hybrid dynamical system is described by*

$$\dot{\boldsymbol{x}} \in \mathcal{F}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{C}, \tag{3.2}$$

$$\boldsymbol{x}^+ \in \mathcal{G}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{D}, \tag{3.3}$$

*where the constrained differential inclusion* (3.2) *constitutes the continuous-time dynamics and the constrained difference inclusion* (3.3) *constitutes the discrete-time dynamics.*

While (3.2)-(3.3) represent the most general formulation for hybrid dynamical systems, many physical systems can be sufficiently described by first-order (differential) equations. In such cases, set-valued mappings and differential inclusions are replaced by equations:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{C}, \tag{3.4}$$

$$\boldsymbol{x}^+ = \boldsymbol{g}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{D}, \tag{3.5}$$

with the respective 4-tuple denoted as

$$\mathcal{H} = (\mathcal{C}, \boldsymbol{f}, \mathcal{D}, \boldsymbol{g})\,. \tag{3.6}$$

The reduced formulation (3.4)-(3.6) is what will be used in the remainder of this thesis. For the sake of an efficient notation, $\mathcal{H}$ and the 4-tuple $(\mathcal{C}, \boldsymbol{f}, \mathcal{D}, \boldsymbol{g})$, hereafter, always refer to the hybrid bouncing ball dynamics defined in Section 3.2.3.

### 3.2.2 Zeno Behavior in Robotic Catching

Independent of how the state of a flying object is determined, uncertainty concerning the true object state remains. Hence, establishing continuous contact between a rigid object and rigid catching device is usually the result of a sequence of (micro) collisions.

**Remark 3.3.** *It is possible to achieve contact without a sequence of collisions when either the object or the catching device allows for permanent deformation, e.g., a sand bag. As a consequence of deformable components, however, it becomes challenging to apply similar amounts of kinetic energy at the throwing stage. Moreover, the assumption of ballistic flight trajectories does not hold for many types of deformable objects.*

One subclass of hybrid systems enabling modeling of collision sequences in mechanical systems is *simple Lagrangian hybrid systems* (SLHS). In [63], SLHS are formally introduced and the conditions for local stability are derived. A property special to hybrid systems, and in particular to SLHS, is the occurrence of an infinite number of collisions in a finite amount of time. In the context of hybrid dynamical systems, the term *Zeno behavior*[1] refers to this property. The presence of a Zeno equilibrium [49] is necessary for the existence of Zeno behavior.

**Definition 3.4.** *Zeno equilibria $\mathcal{A}_\mathrm{Z}$ are points of a hybrid system $\mathcal{H} = (\mathcal{C}, \boldsymbol{f}, \mathcal{D}, \boldsymbol{g})$ for which*

$$\mathcal{A}_\mathrm{Z} = \{\boldsymbol{x} \in \mathcal{D} : \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}) \neq 0\}\,, \tag{3.7}$$

*holds.*

[1] According to Aristotle [77] and Simplicius [89], Zeno of Elea was a pre-Socratic Greek philosopher, who formulated several paradoxes closely related to the described property of hybrid dynamical systems.

Hence, Zeno equilibria are fixed points of the discrete dynamics but are not fixed points of the continuous dynamics. Similar to equilibrium points of traditional dynamical systems, Zeno equilibria can be stable or unstable. Only if Zeno equilibria are stable, can one observe *Zeno behavior*. In the stable case, it is also common to speak of *Zeno solutions* or *Zeno executions* of hybrid systems. Section 3.2.3 introduces the well-studied, one-dimensional hybrid bouncing ball as the SLHS describing the relevant underlying dynamics for catching fast flying objects.

### 3.2.3 Relative Hybrid Bouncing Ball Dynamics in Robotic Catching

In order to analyze the transition into continuous contact, relative system states are defined using the distance definition (2.19) and its time derivative (2.20):

$$\boldsymbol{x} := \begin{bmatrix} h \\ \nu \end{bmatrix}. \tag{3.8}$$

The relative dynamics of the hybrid bouncing ball are described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}) := \begin{bmatrix} x_2 \\ -\gamma(t) \end{bmatrix}. \tag{3.9}$$

These continuous dynamics, however, fail to describe the entire system behavior because collisions with the end-effector surface occur when $h = 0$ in Figure 2.4. For collisions between the object and end-effector evaluated in this thesis, the common Newtonian restitution model is selected based on considerations outlined in Section 2.3.4[2]. This model is an instantaneous damped inversion of the arrival speed described by a coefficient of restitution $\rho \in [0, 1)$. Such restitution constitutes the discrete dynamics

$$\boldsymbol{x}^+ = \boldsymbol{g}(\boldsymbol{x}) := \begin{bmatrix} 0 \\ -\rho x_2 \end{bmatrix}. \tag{3.10}$$

In order to allow a combined stability analysis of the continuous and discrete dynamics, formulate the hybrid bouncing ball dynamics $\mathcal{H} = (\mathcal{C}, \boldsymbol{f}, \mathcal{D}, \boldsymbol{g})$ in the form of (3.4)-(3.6) as

$$\mathcal{C} = \{\boldsymbol{x} : x_1 \geq 0\}, \tag{3.11}$$
$$\mathcal{D} = \{\boldsymbol{x} : x_1 = 0, x_2 \leq 0\}, \tag{3.12}$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{C},\ t \in [0, \infty), \tag{3.13}$$
$$\boldsymbol{x}^+ = \boldsymbol{g}(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \mathcal{D}, \tag{3.14}$$

where $(\mathcal{C}, \boldsymbol{f})$ describe the continuous domain and dynamics (flow) and $(\mathcal{D}, \boldsymbol{g})$ describe the discrete domain and dynamics (jump). Note that this is a non-autonomous system because $\gamma(t)$ is a time-varying function.

[2]The inaccuracy of this model is largely compensated for by the robustness against U3. This robustness is gained by the UZAS notion detailed in Section 3.4.1.

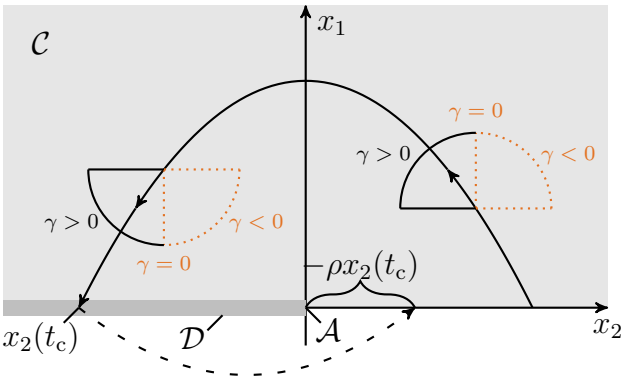


**Figure 3.1:** Flow set (3.11) and jump set (3.12) of the hybrid bouncing ball $\mathcal{H}$ with states $x_1$ as the height of the ball and $x_2$ as its velocity. The cones illustrate the influence of the $\gamma$-sign on the vector field (3.13) in phase space. The dashed line indicates the jump map (3.14).

Figure 3.1 visualizes the flow- and jump-sets described in (3.11) and (3.12). The cones depict the vector field resulting from (3.13) in phase space depending on the sign of $\gamma$. Directions for $\gamma \leq 0$, illustrated by the dotted quadrants of the cones, are prevented hereafter to avoid trajectories never reaching $\mathcal{D}$.[3] For $\gamma > 0$, the system is guaranteed to hit the end-effector at some point, but a time-varying $\gamma$ might still lead to increasing velocities $x_2(t_c)$ between impacts and thus, unstable behavior. Section 3.4 addresses stability for $\gamma > 0$ and its relation to the coefficient of restitution $\rho$.

As revealed by results presented in the remainder of this thesis, success in open-loop nonprehensile catching corresponds to finding accelerations $\gamma(t)$ for $t \geq 0$ that provide asymptotic stability with respect to the compact set

$$\mathcal{A} := \{\boldsymbol{x} : x_1 = 0, x_2 = 0\}\,, \tag{3.15}$$

for a large set of initial relative states. A desirable, asymptotically stabilizing one-dimensional solution $\gamma^*(t)$ can directly be applied to a ballistic catch using the dynamical system parametrization derived in Section 3.3.3. Conditions for $\gamma^*(t)$ to become asymptotically stabilizing are discussed in Section 3.4.

## 3.3 Dimensionality Reduction of Translations in 3D Ballistic Catching

This section presents an approach to close the gap between the well-known hybrid bouncing ball problem from the theory of hybrid dynamical systems and the robotic

[3] Negative values $\gamma \leq 0$ are, however, needed to further reduce impact velocities. The magnitude of the uncertainty in $\boldsymbol{x}$ must be known in this case. Otherwise, relative velocity may change from negative to positive values without a collision, cf. Figure 3.1.

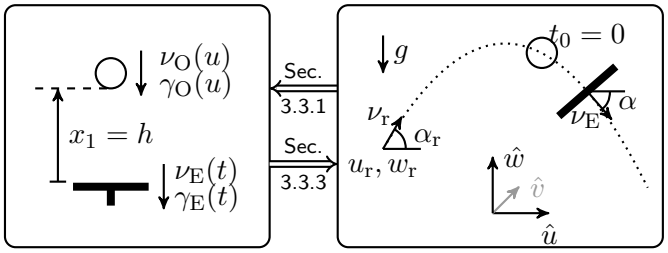


**Figure 3.2: Left:** A hybrid bouncing ball with an accelerated end-effector acting as a unilateral constraint. **Right:** Two-dimensional ballistic trajectory after rotation around $\hat{w}$ such that $\hat{v}$ is normal to the flight plane. The release point is given at time $t = t_r < 0$.

challenge of robustly catching fast-flying objects on their ballistic trajectories. Given that the initial object state is known, Section 3.3.1 first derives a novel parametrization, which formulates the acceleration of the end-effector as input to the robotic catching system. The parametrization builds on the assumption that the object center of mass follows a ballistic trajectory. This assumption applies for rigid objects with a sufficiently large mass-surface ratio such that the influence of aerodynamic drag is small. Hence, given a translational object state at one time instance and despite the uncertainties described in Section 2.3.5, the path on which the object's center of mass travels is predictable and independent of shape. A large range of parts in industrial production processes fulfills this property.

Based on Section 3.3.1, Section 3.3.3 proposes a novel dynamical system motion planner, which requires only three of six states to describe the translational motion: Two positions and the velocity in flight direction instead of three Cartesian positions and velocities each.

## 3.3.1 Flight Path Angle Parametrization

In this section, a parametrization is derived for the flight path angle $\alpha$, which describes the vertical direction in the plane of flight $\Gamma$. Based on the parabolic equations of ballistic flight (2.17), it is shown that the flight path angle $\alpha$ can be expressed as a function of the horizontal position $u$ and a set of constant scenario parameters $\mathcal{P}$ to describe the one-dimensional motion on the ballistic path. The dotted line in Figure 3.2 represents such a ballistic path in $\Gamma$ under the assumption of $\theta = 0$, as discussed in Section 2.3.2. This path is uniquely defined for free object flight if the object's position and velocity vector are known at one time instance, i.e. the initial object state. Therefore, to define the path, the release at $t_r < 0$ with object position $(u_r, w_r)$, object speed $\nu_r$, and object flight path angle $\alpha_r$ are taken as given: $\mathcal{P} := \{t_r, u_r, w_r, \nu_r, \alpha_r\}$. Uncertainties at $t_r$ map to U1 and are, thus, included, whereas the object orientation $\boldsymbol{R}_O(t)$ is neglected during all task planning stages considering Remark 2.2.

The object position for $t > t_\mathrm{r}$ is according to (2.17)

$$u(t) = u_\mathrm{r} + \nu_\mathrm{r}(t - t_\mathrm{r}) \cos \alpha_\mathrm{r}, \tag{3.16}$$
$$w(t) = w_\mathrm{r} + \nu_\mathrm{r}(t - t_\mathrm{r}) \sin \alpha_\mathrm{r} - \tfrac{1}{2} g (t - t_\mathrm{r})^2. \tag{3.17}$$

For the parametrization along the ballistic trajectory, substitute the time of flight by the $u$-coordinate, which identifies the object state uniquely. Solving (3.16) for $t$ gives

$$t(u) = \frac{u - u_\mathrm{r}}{\nu_\mathrm{r} \cos \alpha_\mathrm{r}} + t_\mathrm{r} \ , \tag{3.18}$$

which is only a function of constants and the position $u$. Hence, velocities for a particular object trajectory can be expressed in terms of constants $\mathcal{P}$ and $u$ by

$$\dot{w}(u) = \nu_\mathrm{r} \sin \alpha_\mathrm{r} - g \frac{u - u_\mathrm{r}}{\nu_\mathrm{r} \cos \alpha_\mathrm{r}} \ , \tag{3.19}$$
$$\dot{u} = \nu_\mathrm{r} \cos \alpha_\mathrm{r} \ . \tag{3.20}$$

The flight path angle after release may be denoted by $\alpha(t)$ for $t > t_\mathrm{r}$ or, using (3.18)-(3.20), in the parametrized form

$$\alpha(u) = \tan^{-1}\left(\frac{\dot{w}(u)}{\dot{u}}\right) = \tan^{-1}\left(\tan(\alpha_\mathrm{r}) - g \frac{u - u_\mathrm{r}}{\nu_\mathrm{r}^2 \cos^2 \alpha_\mathrm{r}}\right), \tag{3.21}$$

which results in a negative $\alpha$ for the catching situation in Figure 3.2.

Using the aforementioned parametrization in terms of $u$, the object acceleration in flight direction is given by

$$\gamma_\mathrm{O}(u) = -g \sin\left(\alpha(u)\right). \tag{3.22}$$

The velocity of a free flying object in flight direction (2.13) under assumption (2.16) and parametrization (3.19) is given by

$$\nu_\mathrm{O}(u) = \sqrt{\dot{u}^2 + \left(\dot{w}(u)\right)^2}. \tag{3.23}$$

See also Figure 3.2 for direction and sign conventions.

### 3.3.2 Unicycle Formulation of Free Ballistic Flight

The parametrization derived in Section 3.3.1 enables an efficient description for free ballistic flight in the flight plane $\Gamma$ by the states

$$\boldsymbol{\xi}_\mathrm{O} := \begin{bmatrix} \nu_\mathrm{O} \\ u \\ w \end{bmatrix} = \begin{bmatrix} \nu_\mathrm{O} \\ \bar{\boldsymbol{p}}_\mathrm{O} \end{bmatrix}. \tag{3.24}$$

The state change of an object during free ballistic flight, motivated by a unicycle, is described by the nonlinear dynamical system

$$\dot{\boldsymbol{\xi}}_{\mathrm{O}} = \begin{bmatrix} \gamma_{\mathrm{O}}(\xi_2) \\ \xi_1 \cos\left(\alpha\left(\xi_2\right)\right) \\ \xi_1 \sin\left(\alpha\left(\xi_2\right)\right) \end{bmatrix} = \begin{bmatrix} -g \sin\left(\alpha(\xi_2)\right) \\ \xi_1 \cos\left(\alpha\left(\xi_2\right)\right) \\ \xi_1 \sin\left(\alpha\left(\xi_2\right)\right) \end{bmatrix} \tag{3.25}$$

with $\alpha\left(\xi_2\right)$ from (3.21), which can be solved in real-time by numeric integration subject to an initial state $\boldsymbol{\xi}_{\mathrm{O}}(0) = \begin{bmatrix} \nu_{\mathrm{O}}(0) & u(0) & w(0) \end{bmatrix}^{\mathrm{T}}$ and the constant scenario parameters $\mathcal{P}$. In comparison to (3.25), a unicycle may have one more state such that the change of $\alpha$ becomes an input. Here, however, the parametrized angle $\alpha\left(\xi_2\right)$ reflects the ballistic property.

### 3.3.3 Dynamical System Approach to Ballistic Motion Planning

The focus is, however, on the motion an end-effector $\mathcal{T}^{\mathrm{E}}$ has to perform on the ballistic path in order to stabilize a nonprehensile catch. Therefore, a second pair of variables is defined

$$\bar{\boldsymbol{p}}_{\mathrm{E}}(t) = \begin{bmatrix} u_{\mathrm{E}}(t) \\ w_{\mathrm{E}}(t) \end{bmatrix}, \tag{3.26}$$

and respectively for end-effector velocity $\nu_{\mathrm{E}}$ and acceleration $\gamma_{\mathrm{E}}$. The latter two variables are defined on the ballistic path and are thoroughly discussed in the following. As $\frac{\mathrm{d}}{\mathrm{d}t}\nu_{\mathrm{E}} = \gamma_{\mathrm{E}}$, the acceleration $\gamma_{\mathrm{E}}$ is the input to the robotic catching system and $\nu_{\mathrm{E}}(t_0)$ is an initial speed on the path that needs further discussion.

Having parametrized the ballistic flight path, it remains to derive a motion planner, which can generate the desired end-effector trajectory on the ballistic path in real-time with the end-effector acceleration as input. Therefore, a dynamical system motion generator for P3 is presented to reconstruct an end-effector trajectory

$$\boldsymbol{\xi} := \begin{bmatrix} \nu_{\mathrm{E}} \\ u_{\mathrm{E}} \\ w_{\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \nu_{\mathrm{E}} \\ \bar{\boldsymbol{p}}_{\mathrm{E}} \end{bmatrix} \tag{3.27}$$

that applies solutions $\gamma^*(t)$ to the relative hybrid bouncing ball system $\mathcal{H}$.

A stabilizing relative acceleration $\gamma^*(t)$ on the ballistic path of an object in the $\hat{u}$-$\hat{w}$-plane is tracked with an end-effector motion described by the nonlinear dynamical system

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \gamma_{\mathrm{E}}(t, \xi_2) \\ \xi_1 \cos\left(\alpha\left(\xi_2\right)\right) \\ \xi_1 \sin\left(\alpha\left(\xi_2\right)\right) \end{bmatrix}, \ \boldsymbol{\xi}_0 = \boldsymbol{\xi}(t = 0), \tag{3.28}$$

with acceleration $\gamma_{\mathrm{E}}(t, \xi_2) = \gamma_{\mathrm{O}}(\xi_2) - \gamma^*(t)$ and $\alpha$ from (3.21) evaluated at $\xi_2 = u_{\mathrm{E}}$. Hereby, the first state $\xi_1 = \nu_{\mathrm{E}}$ is the one-dimensional end-effector velocity on the

ballistic path, whereas the other two states determine the position of the end-effector in the plane of flight, cf. Figure 3.2.

The initial end-effector state is defined

$$\boldsymbol{\xi}_0 = \begin{bmatrix} \nu_{\mathrm{O}}(0) + x_2(0) \\ u(0) + (x_1(0) + r)\cos(\alpha(0)) \\ w(0) + (x_1(0) + r)\sin(\alpha(0)) \end{bmatrix} \tag{3.29}$$

using (3.16), (3.17), (3.21) and (3.23). The scalar $r$ denotes the radius of the object circumcircle and therefore ensures that object and end-effector are initially disjoint.[4] Note also that (3.29) is based on the assumption of sufficiently small $(x_1(0)+r)$, which allows to assume $\gamma_{\mathrm{O}}(u_{\mathrm{E}}) \cong \gamma_{\mathrm{O}}(u)$ in the remainder of this article. For a given initial end-effector state $\boldsymbol{\xi}_0$, the desired end-effector trajectory $\boldsymbol{\xi}(t)$ thus results from a numeric integration of (3.28) with input $\gamma^*(t)$.

Section 3.4 analyzes and discusses the influence of $\gamma$ on bouncing ball dynamics and thus on robotic catching. The choice of the relative initial state $\boldsymbol{x}(0)$ is crucial for the success of an open-loop catch and will be discussed in Section 3.4.5. As a result, Section 4.3.2 proposes an optimization-based motion planner for robotic catching that finds $\gamma^*(t)$ while taking dynamic limitations of the robot into account.

## 3.4 Uniform Zeno Asymptotic Stability for Robotic Catching

This section reviews the notion of uniform Zeno asymptotic stability (UZAS, [64]) in Section 3.4.1, including its potential for robotic catching. Enabled by UZAS, Section 3.4.6 proposes a novel, provable, upper bound on the maximal remaining rebound height for bouncing ball systems that fulfill the UZAS property. Upon fulfillment of the UZAS property, the notion of an exact tight bound on the maximal Zeno time from [64] also becomes applicable, as presented in Section 3.4.2. Starting from the bound on the maximal Zeno time, Section 3.4.4 proves robustness for a bounded set of uncertain restitution behavior. In the presence of provably stabilizable Zeno dynamics, Section 3.4.3 reduces the general catching problem formulation with its infinite time horizon to a special catching task problem formulation with a finite time horizon. The last extension of maximal Zeno time notion is derived in Section 3.4.5 and encompasses the initial relative velocity between the object and catching device, which potentially maximizes the robustness against initial distance uncertainties. A realistic numeric simulation motivated by the experiments presented in Chapter 5 and a summarizing discussion is provided at the end of this section.

[4] If a range of differently sized objects should be caught with the same setup, choose $r$ from the largest possible object at the cost of increasing U1.

### 3.4.1 Uniform Zeno Asymptotic Stability (UZAS)

Consider the hybrid dynamical system $\mathcal{H}$ from (3.11)–(3.14) with set-valued acceleration

$$\gamma(t) \in [\gamma_{\min}, \gamma_{\max}], \text{ where } 0 < \gamma_{\min} \leq \gamma_{\max}. \tag{3.30}$$

Or *et al.* then proved the following theorem in [64].

**Theorem 3.5** (Or *et al.* [64, Th. 1])**.** *The origin of a bouncing ball $\mathcal{H}$ with set-valued acceleration* (3.30)*, possesses uniform Zeno stability if and only if*

$$\rho^2 < \frac{\gamma_{\min}}{\gamma_{\max}} \tag{3.31}$$

*holds.*

The detailed proof can be found in [64], whereas parts important for this thesis are highlighted in the following. As a result, a novel view on the robustness problem, formulated with U1-U3, becomes available and will turn out practical for quantizing robustness in object catching.

The proof relies on a sophisticated Lyapunov function $V : \mathcal{U} \to \mathbb{R}_{\geq 0}$ with $\mathcal{U} = \{\boldsymbol{x} \in \mathbb{R}^2 : W(\boldsymbol{x}) > 0\}$ defined by

$$\begin{aligned} &V(\boldsymbol{x}) = \kappa x_2 + \sqrt{W(\boldsymbol{x})}, \text{ where } W(\boldsymbol{x}) = \frac{1}{2p(x_2)} x_2^2 + x_1, \\ &\text{with } p(x_2) = \begin{cases} \gamma_{\max} & \text{if } x_2 \leq 0 \\ \gamma_{\min} & \text{if } x_2 > 0 \end{cases} \\ &\text{and } \kappa = \left( \frac{1}{\sqrt{2\gamma_{\max}}} - \frac{\rho}{\sqrt{2\gamma_{\min}}} \right) \frac{1}{1+\rho}. \end{aligned} \tag{3.32}$$

Interestingly this piecewise defined Lyapunov function is continuous and continuously differentiable on $\operatorname{dom} V$ even though kinetic energy is deducted instantaneously from the hybrid bouncing ball system at every collision:

$$V(g(\boldsymbol{x})) = V(\boldsymbol{x}) \tag{3.33}$$

for all $\boldsymbol{x} \in \mathcal{D}$. Furthermore, by verifying that

$$\langle \nabla V(\boldsymbol{x}), \boldsymbol{f} \rangle \leq -\kappa \gamma_{\min} \tag{3.34}$$

holds for all $\boldsymbol{x} \in \mathcal{C} \backslash \{0\}$, one concludes that $V(\boldsymbol{x})$ is a strictly decreasing Lyapunov function. Therefore the origin is uniform Zeno asymptotically stable (UZAS).

For the bouncing ball follows from (3.30) robustness with respect to uncertain relative acceleration (see U2) as visualized with single flow periods in Figure 3.3. From (3.31) follows robustness with respect to uncertain restitution behavior (see U3). Furthermore, it becomes possible to establish continuous contact with a single open-loop acceleration $\gamma(t)$ for a range of initial states $\boldsymbol{x}(0) \in \mathcal{C} \cup \mathcal{D}$, which relates to U1. Hereby, $\gamma(t)$ might even be chosen constant. As a result, given a sufficiently small U1 provided by the repeatable robot throw, online replanning of catching motions might become obsolete as well as real-time measurements of the object state.

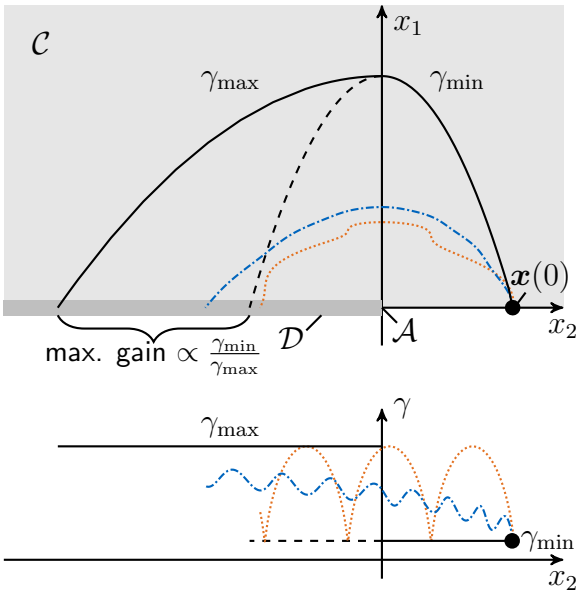


**Figure 3.3:** Flow and jump sets of the hybrid bouncing ball $\mathcal{H}$ over a single period starting at $\boldsymbol{x}(0)$. The acceleration $\gamma$ satisfies (3.30) and (3.31). Any fast or slowly varying acceleration within the bounded set leads to a lower rebound height than a single switch from the lower to the upper bound at the maximum height ($\gamma = -p(x_2)$), which is the "most unstable solution" [64].

### 3.4.2 Maximal Zeno Time

The UZAS condition (3.31) from Theorem 3.5 formulates cases, where the general catching goal (2.22) can be achieved as $t \to \infty$ and therefore where a finite Zeno time $Z$ exists. Consequently, the notion of a maximal Zeno time [64] at particular time $t'$ becomes also applicable, whereas the abbreviations for the relative height $x_1(t') =: h'$ and relative velocity $x_2(t') =: \nu'$ are used.

**Theorem 3.6** (Or *et al.* [64, Th. 2])**.** *If Theorem 3.5 holds, all solutions are Zeno and their maximal Zeno time is given by*

$$Z_{\max}(h', \nu') = \begin{cases} \frac{\nu' + \sigma U_{0,\min}}{\gamma_{\min}} & \text{for} \quad \nu' \geq \nu_c \\ \frac{\nu' + U_{0,\max}(1+\beta\delta)}{\gamma_{\max}} & \text{for} \quad \nu' < \nu_c \end{cases} \tag{3.35}$$

*where*

$$\begin{aligned} U_{0,\max} &= \sqrt{(\nu')^2 + 2\gamma_{\max} h'} \\ U_{0,\min} &= \sqrt{(\nu')^2 + 2\gamma_{\min} h'} \\ \nu_c &= -\sqrt{\frac{2\gamma_{\min} h'}{\sigma^2 - 1}}, \quad \delta = \frac{\gamma_{\max}}{\gamma_{\min}} \\ \beta &= 2\rho \frac{1+\rho}{1-\rho^2\delta}, \quad \sigma = \sqrt{1 + 2\beta + \delta\beta^2}. \end{aligned} \tag{3.36}$$

*Proof.* See proof in [64, Th. 2]. □

An upper bound on the Zeno time for bouncing ball systems was also discussed before the aforementioned referenced contribution. In [37], a more conservative bound was derived, which was, shown in [64] to be never a tight bound. In contrast, Theorem 3.6 states a provably exact tight bound on the Zeno time of the set-valued bouncing ball under any initial condition.

### 3.4.3 Special Catching Task Problem Formulation

In practice, the general formulation (2.22) with $t \to \infty$, evaluated at $t = 0\,\mathrm{s}$, requires consideration of acceleration changes in phases P3 and P4, which leads to large $\delta$-values. Sizeable values for $Z_{\max}$ and small valid ranges for $\rho$ result. After convergence, however, the relative acceleration must only be negative ($\gamma > 0$) to maintain contact. Here, it is therefore proposed to limit the time horizon to the tracking phase P3 with controllable relative acceleration. By accounting for the finite convergence time property of Zeno behavior the following conjecture reformulates the general catching goal (2.22).

**Conjecture 3.7.** *Subject to a deceleration phase that ensures to maintain continuous contact with an object that has been established earlier, a catch is considered successful if, and only if,*

$$h(t \to t_\mathrm{f}) = 0 \quad \text{and} \quad \nu(t \to t_\mathrm{f}) = 0 \tag{3.37}$$

*is fulfilled.*

Hence, successful catching becomes predictable for initial relative states that fulfill $Z_{\max}(\boldsymbol{x}(0)) < t_\mathrm{f}$, also subject to U1-U3.

### 3.4.4 Robustness Against Uncertain Restitution Behavior

The tight bound on the maximal Zeno time (3.35) could also be solved for $h'(Z_{\max}, \nu')$. Assuming a constant $\bar{Z}_{\max}$, a piecewise quadratic parabola results, which encloses a convex area

$$\mathcal{X}_{\bar{Z}_{\max}} := \left\{ \boldsymbol{x} \in \mathcal{C} \cap \mathcal{D} | Z_{\max} < \bar{Z}_{\max} \right\} \tag{3.38}$$

with the ($h' = 0$)-axis of the state space. Moreover, the maximal Zeno time (3.35) depends on the coefficient of restitution $\rho$. In order to show that a set $\mathcal{X}_{\bar{Z}_{\max}}$, found for a particular $\rho$, also applies for all lower values of $\rho$, the following Lemma and Corollary prove that $Z_{\max}$ strictly increases as $\rho \in [0, 1)$ increases.

**Lemma 3.8.** *If Theorem 3.5 is fulfilled, the function*

$$\beta = 2\rho \frac{1+\rho}{1-\rho^2\delta} \tag{3.39}$$

*from* (3.36) *with* $\delta = \frac{\gamma_{\max}}{\gamma_{\min}}$ *is positive and strictly increasing as* $\rho \in [0, 1)$ *increases.*

*Proof.* In order to proof the Lemma, the inequality $\frac{\partial \beta}{\partial \rho} > 0$ must be shown to hold in the domain of $\rho$. Therefore, consider the derivative of (3.39) with respect to $\rho$:

$$\frac{\partial \beta}{\partial \rho} = \frac{2\left(\delta\rho^2 + 2\rho + 1\right)}{\left(1-\delta\rho^2\right)^2}. \tag{3.40}$$

Since Theorem 3.5 is fulfilled, it is $0 \leq \delta\rho^2 < 1$. Hence, the inequality $\frac{\partial \beta}{\partial \rho} > 0$ and $1 - \rho^2\delta > 0$ in the denominator of (3.39) always holds. □

With the help of Lemma 3.8, the following Corollary can be derived from Theorem 3.6.

**Corollary 3.9.** *If Theorem 3.5 is fulfilled, the tight bound on the Zeno time* $Z_{\max}$ *from* (3.35)-(3.36) *strictly increases as* $\rho \in [0, 1)$ *increases.*

*Proof.* Considering that the multiplication of two positive strictly increasing functions results again in a positive strictly increasing function and in view of Lemma 3.8, it is sufficient to show that $Z_{\max} > 0$ and $\frac{\partial}{\partial \beta} Z_{\max} > 0$ hold. The earlier is true for (3.35) because due to $U_{0,\min} > |\nu'|$, $U_{0,\max} > |\nu'|$, $\sigma > 1$, and $\delta > 1$ it is

$$\sigma U_{0,\min} \geq |\nu'| \quad \text{and} \tag{3.41}$$

$$U_{0,\max}(1 + \beta\delta) \geq |\nu'| \tag{3.42}$$

in the designated domain. Moreover, the derivatives compute as

$$\frac{\partial}{\partial \beta} Z_{\max} = \begin{cases} \frac{(2+2\delta\beta)U_{0,\min}}{\gamma_{\min}\sigma} & \text{for} \quad \nu' \geq \nu_c \\ \frac{\delta U_{0,\max}}{\gamma_{\max}} & \text{for} \quad \nu' < \nu_c \end{cases}. \tag{3.43}$$

Hence, the second condition $\frac{\partial}{\partial \beta} Z_{\max} > 0$ also holds, which completes the proof. □

The physical interpretation of (3.38) and Corollary 3.9 in the context of catching is as follows. If successful catching can be guaranteed with a particular restitution $\rho$ and catching motion, then this guarantee also applies to all lower restitutions under the same motion. As a result, objects of various materials can be caught simultaneously with a single strategy and, therefore, without re-planning. The resulting potential to overcome the problem of difficult and imprecise estimation of an object-dependent restitution behavior is even more important. Only a worst-case bound of the coefficient of restitution must be known or assumed in a sufficiently conservative way.

### 3.4.5 Maximized Distance Uncertainty Compensation

Following the observation that $\mathcal{X}_{\bar{Z}_{\max}}$ is enclosed by a downwardly opened, piecewise quadratic parabola intersecting the $(h' = 0)$-axis, the question arises as to what initial relative state $\boldsymbol{x}^{\mathrm{rob}}(0)$ in $\mathcal{X}_{\bar{Z}_{\max}}$ maximizes catching robustness. The best option for a nominal initial state, however, depends on the expected magnitude in each dimension of U1. Nonetheless, the following corollary provides an orientation for potentially effective nominal initial relative velocities. Generally speaking, the following corollary returns the velocity $x_2(0)$ for which the piecewise quadratic parabola reaches the maximal $x_1(0)$-value.

**Corollary 3.10.** *Assume condition* (3.31) *is satisfied, then the nominal initial relative velocity*

$$x_2^{\mathrm{rob}}(0) = -\frac{\gamma_{\min} Z_{\max}}{2\beta + \delta\beta^2}, \tag{3.44}$$

*with*

$$\begin{aligned} \delta &= \frac{\gamma_{\max}}{\gamma_{\min}} \\ \beta &= 2\rho \frac{1+\rho}{1-\rho^2\delta}, \end{aligned} \tag{3.45}$$

*from* (3.36) *maximizes the range of* $x_1(0)$ *for which* (3.37) *holds,* $Z_{\max} = t_{\mathrm{f}} - t_0$ *is the duration of P3 and* $\beta$ *is calculated based on the maximum expected restitution in* $h$*-direction.*

*Proof.* First, rearrange the first case in (3.35) to become a function of $Z_{\max}$ and $\nu$ given by

$$h = \frac{(1-\sigma^2)\nu^2 - 2\gamma_{\min} Z_{\max}\nu + (\gamma_{\min} Z_{\max})^2}{2\sigma^2\gamma_{\min}}. \tag{3.46}$$

Then, take the derivative

$$\frac{\mathrm{d}h}{\mathrm{d}\nu} = \frac{(1-\sigma^2)\nu - \gamma_{\min} Z_{\max}}{\sigma^2\gamma_{\min}}, \tag{3.47}$$

where setting (3.47) equal to zero results in

$$\nu = \frac{\gamma_{\min} Z_{\max}}{1-\sigma^2} = -\frac{\gamma_{\min} Z_{\max}}{2\beta + \delta\beta^2}. \tag{3.48}$$

Considering that $\sigma^2 > 1$, a second derivative $\frac{\mathrm{d}^2 h}{\mathrm{d}\nu^2}$ will always be negative and thus (3.48) is a maximum.[5] Steps (3.46)-(3.48) repeated for the second case in (3.35) result in

$$h = \frac{(1-(1+\beta\delta)^2)\nu^2 - 2\gamma_{\max} Z_{\max}\nu + (\gamma_{\max} Z_{\max})^2}{2\left(1+\beta\delta\right)^2 \gamma_{\max}}, \tag{3.49}$$

again taking the derivative

$$\frac{\mathrm{d}h}{\mathrm{d}\nu} = \frac{-\left(2\beta\delta + \beta^2\delta^2\right)\nu - \gamma_{\max} Z_{\max}}{\left(1+\beta\delta\right)^2 \gamma_{\max}}. \tag{3.50}$$

Then, replace $\gamma_{\max} = \delta\gamma_{\min}$ leads to

$$\frac{\mathrm{d}h}{\mathrm{d}\nu} = \frac{-(2\beta + \beta^2\delta)\nu - \gamma_{\min} Z_{\max}}{\left(1+\beta\delta\right)^2 \gamma_{\min}}. \tag{3.51}$$

Finally, setting (3.51) equal to zero results again in

$$\nu = -\frac{\gamma_{\min} Z_{\max}}{2\beta + \delta\beta^2} = \frac{\gamma_{\min} Z_{\max}}{1-\sigma^2}, \tag{3.52}$$

which completes the proof. □

As a result, a relative velocity according to Corollary 3.10 for motion planning of future catching robots is recommended, as the knowledge of object states is inevitably imprecise (U1). Hence, a negative offset for $x_2(0)$, besides providing the necessary negative relative acceleration ($\gamma > 0$), is suggested as a primary measure to increase success in robotic catching, as well as for existing robots. If reliable success is achieved, one may further reduce impact velocities with velocity matching $x_2(0) \to 0$, as considered in previous dynamic catching approaches of a lower success rate.

### 3.4.6 Maximal Rebound Height and Limited End-Effector Domain

Besides the stability of system $\mathcal{H}$, knowing a bound on the rebound height after an arbitrary point in time $t'$ subject to (3.30)–(3.31) is useful in the process of predicting success or failure in robotic catching. The following corollary offers an easily applicable answer.

[5] Note that (3.48) is directly related to $\nu_c$ in (3.36), e.g., by inserting the first case of (3.35) into (3.48) and resolving for $\nu$.

**Corollary 3.11.** *If the acceleration $\gamma(t)$ of the hybrid bouncing ball system $\mathcal{H}$ satisfies* (3.30) *and* (3.31) *for all $t > t'$, then the maximum possible rebound height for all $t > t'$ is limited by*

$$\bar{x}_1(h', \nu') = \begin{cases} h' + \frac{(\nu')^2}{2\gamma_{\min}} & \text{for} \quad \nu' \geq 0 \\ \max\left[h', \frac{\rho^2\left((\nu')^2 + 2\gamma_{\max} h'\right)}{2\gamma_{\min}}\right] & \text{for} \quad \nu' < 0 \end{cases} \tag{3.53}$$

*based on the relative state $\boldsymbol{x}(t') =: \begin{bmatrix} h' & \nu' \end{bmatrix}^{\mathrm{T}}$.*

*Proof.* Due to (3.30), the ball reaches the highest point at zero relative velocity ($x_2 = 0$) in every rebounding cycle. Furthermore, due to the UZAS property, in every consecutive cycle the highest point $x_1(x_2 = 0)$ is smaller than in the previous one. Hence, in order to compute $\bar{x}_1$, it is sufficient to calculate the first $x_1(x_2 = 0)$ after $t'$ in the presence of the worst-case acceleration scenario given by $p(x_2)$ in (3.32).

The first case in (3.53) represents the rising phase, i.e. $\nu' \geq 0$. Here, simply the current height $h'$ is added to the height that could be gained by transforming the current kinetic energy into potential energy under the smallest possible acceleration.

In the falling case $\nu' < 0$, a collision occurs before the next peak height is reached. Denoting the first post-impact velocity after $t'$ as $\nu^+$, the gain of height is already know from the previous step with $h' = 0$ as

$$\bar{x}_1 = \frac{(\nu^+)^2}{2\gamma_{\min}}. \tag{3.54}$$

With the restitution law $\nu^+ = -\rho\nu^-$ the relation to the pre-impact state becomes

$$\bar{x}_1 = \frac{(-\rho\nu^-)^2}{2\gamma_{\min}}. \tag{3.55}$$

Finally, the maximum possible velocity of $\nu^-$ is the kinetic energy of $\nu'$ in addition to the velocity gained by transforming $h'$ into kinetic energy under the largest possible acceleration

$$\nu^- = -\sqrt{(\nu')^2 + 2\gamma_{\max} h'}. \tag{3.56}$$

Inserting (3.56) in (3.55) results in the second line of (3.53), considering that $\bar{x}_1$ must never be smaller than the current height $h'$. □

For acceleration ratios (3.31) close to one, Theorem 3.5 and Corollary 3.11 are not restrictive because nearly the full range of the coefficient of restitution $\rho$ is allowed. Vice versa, the restrictions on $\rho$ become tight if a large range of accelerations $\gamma$ must be covered.

In nonprehensile robotic catching, it will later be shown that convergence speed improves as $\gamma_{\min}$ increases, which can also be concluded from (3.34). Large acceleration, however, may only be provided in P3 (tracking) because P4 (deceleration) is typically

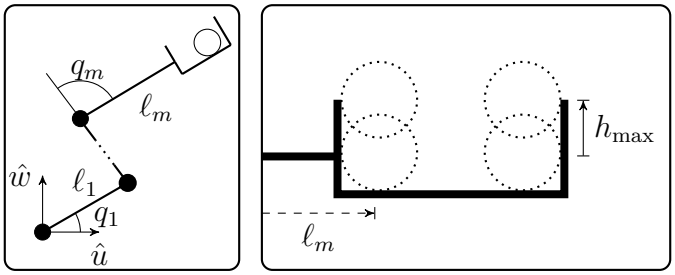


**Figure 3.4: Left:** Kinematic model for $m$ actuated rotational degrees of freedom. The $\hat{u}$-$\hat{w}$-frame coincides with the robot base. **Right:** Unactuated, box-like end-effector with height $h_{\max}$.

governed by comparably little acceleration. Under typical conditions the Zeno time may lie in P4, illustrated with a numerical example in Section 3.4.7.Due to the small lower acceleration bound $\gamma_{\min}^{\mathrm{P4}}$ a restrictively large range of accelerations may apply.

So far, the end-effector has been assumed planar like a plate. However, even if (3.37) holds and no form or force closure is used, the object may start to roll or slide on the end-effector after entering P4. Therefore, a box-like end-effector design as depicted in Figure 3.4 is proposed to prevent objects from falling off. Moreover, such motions parallel to $\mathcal{S}_{\mathrm{E}}$ have an additive or subtractive effect (depending on the direction) on the relative acceleration between object and end-effector, see Section 3.5. Contact with the box walls may even induce additional velocity. In order to prevent the object from losing contact with $\mathcal{S}_{\mathrm{E}}$, motion planning in P4 must pursue sufficiently large negative relative acceleration.

In case of spherical objects and in view of Corollary 3.11, a box-like end-effector may even enable successful catching if

$$\bar{x}_1(h', \nu') < h_{\max} \quad \text{at} \quad t' = t_{\mathrm{f}}, \tag{3.57}$$

holds, whereas $h_{\max}$ denotes the box height reduced by the object radius.

### 3.4.7 Numerical Example

Consider a relative initial state $\boldsymbol{x}(0) = \begin{bmatrix} 0 & 0.9 \end{bmatrix}^{\mathrm{T}}$ at start of P3, which means the hybrid bouncing ball $\mathcal{H}$ defined in (3.11)–(3.14) is initially in contact and has positive velocity. The restitution is chosen $\rho = 0.35$. In P3, the robot moves according to (3.28) with the goal to apply $\gamma^* = 25\,\mathrm{ms}^{-2}$. In P4, the goal is to maintain previously established contact with $\gamma > 4\,\mathrm{ms}^{-2}$. Two acceleration patterns are simulated to represent U2 in the presence of these goals. Both are depicted in the lower plot of Figure 3.5. The solid line shows a simulation of noisy acceleration by means of a high frequent sine function between $20\,\mathrm{ms}^{-2}$ and $30\,\mathrm{ms}^{-2}$ in P3 and between $4\,\mathrm{ms}^{-2}$ and $20\,\mathrm{ms}^{-2}$ in P4. The dotted line depicts a simulation of the "most unstable acceleration" [64] that is $\gamma = -p(x_2)$ using the previously mentioned bounds.

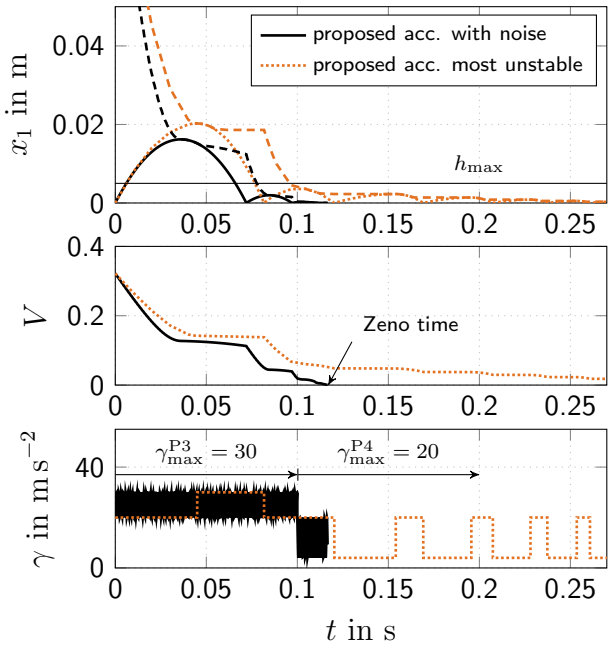


**Figure 3.5:** Height of a bouncing ball over time with initial state $\boldsymbol{x}(0) = [0 \quad 0.9]^{\mathrm{T}}$ and restitution $\rho = 0.35$. The dashed lines in the upper plot indicate the respective upper bound $\bar{x}_1$ from (3.53), which is decreasing. The Lyapunov function (middle) based on (3.32) is continuous and decreasing to zero in finite (Zeno) time. Dynamics after the Zeno time are not considered. The upper acceleration bound $\gamma_{\max}$ drops at $t_{\mathrm{f}} = 0.1\,\mathrm{s}$, which accounts for limited robot tracking capabilities.

The first two plots of Figure 3.5 show the simulation result for height and Lyapunov function value (3.32) over time for a sequence of impacts. The fast variations of $\gamma$, which intend to simulate noise, are not visible in the two upper plots, which is due to the low pass property of the double integrating flow. Moreover, the Lyapunov function value decreases faster during $t < 0.1\,\mathrm{s}$, which is in line with (3.34) and thus confirms that large $\gamma_{\min}$ is desirable.

In addition, the first plot also displays the evaluation of Corollary 3.11 with dashed lines. For this evaluation, uncertainty U2 and the drop in relative acceleration at $t = 0.1\,\mathrm{s}$ is taken into account by $\gamma_{\max}^{\mathrm{P3}} = 30\,\mathrm{ms}^{-2}$ and $\gamma_{\min}^{\mathrm{P3}} = 4\,\mathrm{ms}^{-2}$, which reduces to $\gamma_{\max}^{\mathrm{P4}} = 20\,\mathrm{ms}^{-2}$ and $\gamma_{\min}^{\mathrm{P4}} = 4\,\mathrm{ms}^{-2}$ for $t > 0.1\,\mathrm{s}$. At the example of a maximum rebound height $h_{\max} = 5\,\mathrm{mm}$, indicated by the horizontal line in the first plot of Figure 3.5, one can see that (3.53) already holds from $t = 0.1\,\mathrm{s}$. The simulated Zeno time here is at $t = 0.34\,\mathrm{s}$ in the most unstable case. Limiting the analysis to only those solutions that have Zeno times that lie in P3 might thus be overly restrictive.

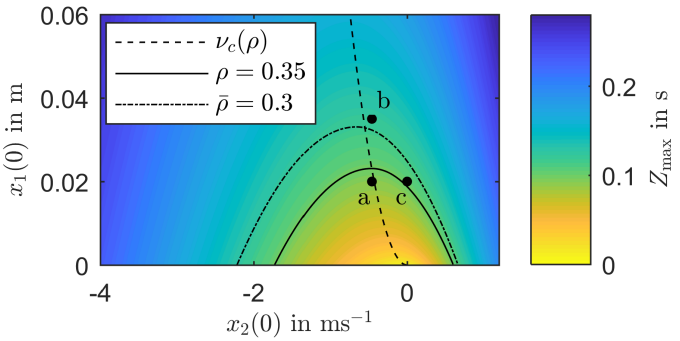


**Figure 3.6:** Contours indicating the maximal Zeno time (3.35) for Example 1 acceleration bounds in P3 $\left(\gamma(t) \in [20, 30]\ \text{ms}^{-2}\right)$ subject to the initial relative state $\boldsymbol{x}' = \boldsymbol{x}(0)$ and $\rho = 0.35$. All initial states below the solid contour fulfill $Z_{\max} < t_{\text{f}} = 0.1\,\text{s}$. The dash-dotted contour indicates the trend of the $Z_{\max} < t_{\text{f}}$ contour as $\rho$ decreases.

### Maximal Zeno Time

For the above example, Figure 3.6 visualizes the maximal possible Zeno time (3.35) at the start of P3 ($t_0 = 0$) depending on the initial relative state $\boldsymbol{x}(0)$. The solid contour indicates $Z_{\max} = 0.1\,\text{s}$, which allows to conclude that all initial relative states enclosed by this contour and the horizontal axis fulfill (3.37), and thus lead to continuous contact already in P3. The dash-dotted contour indicates the effect of U3, which illustrates that (3.35) is monotonically decreasing as $\rho$ decreases as proven with Corollary 3.9. Vice versa, the admissible U1 fulfilling $Z_{\max} < t_{\text{f}}$ increases.

### Maximal Remaining Rebound Height

Consider again the aforementioned example with restitution $\rho = 0.35$ and the same range of potential initial relative states as in Figure 3.6. A simulation now calculates how the relative object state $\boldsymbol{x}$ evolves until $t_{\text{f}} = 100\,\text{ms}$ based on the "most unstable" acceleration pattern illustrated in Figure 3.5. At time $t = t_{\text{f}}$, Corollary 3.11 is evaluated for all $\boldsymbol{x}(t_{\text{f}}) = \begin{bmatrix} h_{\text{f}} & \nu_{\text{f}} \end{bmatrix}^{\text{T}}$ using $\gamma_{\max}^{\text{P4}} = 20\,\text{ms}^{-2}$ and $\gamma_{\min}^{\text{P4}} = 4\,\text{ms}^{-2}$. Figure 3.7 illustrates the results.

Similar to the previous maximal Zeno time example, negative relative velocities turn out advantageous, as rebounds remain comparably small with respect to distance uncertainties.

## 3.4.8 Discussion

This section reviews uniform Zeno asymptotic stability (UZAS) and provides a basis for extensions that lead to provable robustness in catching. Here, the robustness for

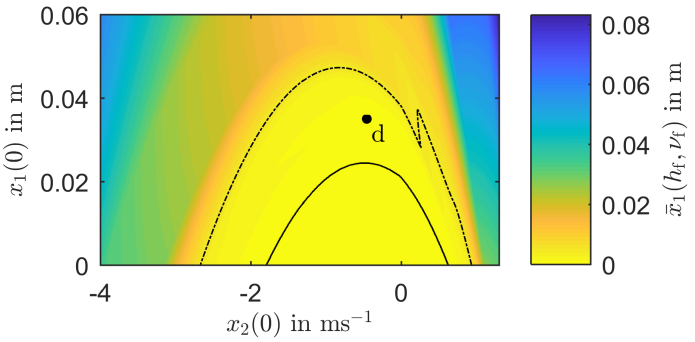


**Figure 3.7:** Contours indicating the maximal possible rebound height (3.53) for the acceleration bounds $\big(\gamma(t) \in [20, 30]\ \mathrm{ms}^{-2}\big)$ at the end of P3 ($t' = t_\mathrm{f}$) and subject to $\rho = 0.35$. All initial states below the solid contour fulfill $\bar{x}_1(h_\mathrm{f}, \nu_\mathrm{f}) = 0\,\mathrm{mm}$ (Zeno solutions). The dash-dotted contour depicts the $\bar{x}_1(h_\mathrm{f}, \nu_\mathrm{f}) < 5\,\mathrm{mm}$ contour.

catching originates in the set-valued conditions that must hold for maintaining UZAS. Extensions such as the tight bound on the maximal Zeno time or estimation of the maximal remaining rebound height rely on UZAS and, therefore, benefit from the generality provided by the set-valued formulation. The proposed maximal distance uncertainty compensation further and actively exploits the maximal Zeno time. These extensions are particularly important for robotic catching because the time a robot is capable of tracking a ballistic flight trajectory is typically very brief. Hence, this section proposes replacing the global problem of successful catching with the goal of establishing continuous contact before the tracking phase ends, subject to mild requirements for the subsequent deceleration phase. The set of initial relative states from which continuous contact could be established in the tracking phase is, however, relatively small. Therefore, as a second approach, the notion of the maximal remaining rebound height is introduced. Together with the proposition of a box-shaped end-effector, success also becomes possible when small rebounds remain at the end of the tracking phase.

The two most important properties of the UZAS conditions are the formulation for set-valued relative acceleration (3.30) and the inequality condition (3.31), which relates the coefficient of restitution with the acceleration bounds. UZAS regarding set-valued acceleration enables mapping or compensating for a variety of uncertainties in a generalized way, for example:

- inaccurate robot motions,
- erroneous prediction of the object acceleration in the ballistic scenario due to inaccurate estimation of the initial object state, e.g., by visual perception systems or a throwing robot, and

- negligence of the end-effector orientation during motion planning, which results in motions perpendicular to the ballistic flight trajectory after impact (see also Section 3.5).

The individual identification of these exemplary sources of uncertainty is time-consuming and expensive.

The inequality condition (3.31) for UZAS allows for the use of a simple Newtonian restitution model (2.25) without losing generality as UZAS holds for all $\rho \in \left[0 \quad \sqrt{\frac{\gamma_{\min}}{\gamma_{\max}}}\right)$. This set-valued condition for the coefficient of restitution is of a generalizing nature because it maps to a range of objects (material and shape) without requiring precise restitution models for each combination of object and end-effector. Moreover, the generalized range of objects for which UZAS holds is inversely proportional to the acceleration uncertainty. In the extreme case of no acceleration uncertainty in the relative system between an object and unilateral constraint, any collision not inducing energy in the direction of flight results in UZAS. Inversely, substantial acceleration uncertainty requires sizable energy dissipation at collisions (small $\rho$) to guarantee UZAS. Given this discussion on UZAS, robust catching no longer requires expert knowledge in the large field of modeling impact dynamics, which is known for being sensitive regarding inaccurate modeling. Only impacts that induce energy in the direction perpendicular to the unilateral constraint ($\rho > 1$) might not be rigorously covered by the theory presented in this section. Nevertheless, increasing the robustness against U1-U3 generally considering (3.30) and (3.31) can also be expected to increase the robustness if more complex objects are to be caught.

Furthermore, UZAS is proven using a sophisticated Lyapunov function that is always continuous, even when velocity resets at collisions. The flow condition (3.34) in the proof should be afforded special attention, as it strictly requires the Lyapunov function to be decreasing. The upper bound $-\kappa\gamma_{\min}$ on the right-hand side of the inequality only depends on the upper and lower acceleration bounds and the coefficient of restitution. The physical conclusion from (3.34), beyond its role in the proof, is that catching robustness generally increases as $\gamma_{\min}$ increases. Moreover, this upper bound on the decrease of the Lyapunov function directly relates to the finite convergence time property of UZAS. Chapter 5 considers this observation and demonstrates how it can be used for robustification in robotic catching systems.

As UZAS is enabled for ballistic robotic catching (see Section 3.3.3), the maximal Zeno time notion from [64] also becomes applicable. The analytic solution for calculating the maximal Zeno time reviewed in Theorem 3.6 states an exact tight bound given a coefficient of restitution, acceleration bounds and an initial relative object state. In contrast, traditional stability and convergence notions typically state if and how fast the system states converge to a desired value as $t \to \infty$. Alternatively, the finite time until a state is guaranteed to remain within a certain proximity to a desired value is taken as a measure. Thus, both of these traditional approaches are ill-suited for describing the finite time problem of establishing continuous contact. The main disadvantage of

the infinite time horizon formulation is that not only the acceleration bound from the tracking phase (P3), but also from the deceleration phase (P4) must be considered. The latter (P4) is characterized by a much lower value $\gamma_{\min}$ than in P3. Hence, a union of the acceleration bounds from P3 and P4 covers large acceleration uncertainties resulting in small $\frac{\gamma_{\min}}{\gamma_{\max}}$ and, therefore, large $Z_{\max}$. Enabled by the explicit consideration of Zeno behavior and motivated by these observations, the proposed special catching task formulation enables the reduction of the success quantification to the controllable tracking phase. In this context, the condition $Z_{\max} \leq t_{\mathrm{f}}$ is considered as a success indicator.

As a result, a convex set $\mathcal{X}_{\bar{Z}_{\max}}$ can be implicitly formulated collecting all states guaranteed to have a maximal Zeno time less than or equal to $\bar{Z}_{\max} := t_{\mathrm{f}}$, subject to a particular restitution $\rho$. Considering the typically short tracking phases (P3) in robotic catching, this formulation is later demonstrated useful in quantifying the robustness of catching systems and, therefore, making these systems comparable for the first time. The interpretation of $\mathcal{X}_{\bar{Z}_{\max}}$ is extended with Corollary 3.9 . By proving that the maximal Zeno time $Z_{\max}$ strictly increases as the coefficient of restitution $\rho$ increases, $\mathcal{X}_{\bar{Z}_{\max}}$ also becomes valid for all $\rho$ below a reference value. The resulting tolerance regarding the coefficient of restitution is crucial for practical catching situations in which identifying impact models is expensive and error-prone.

Nonetheless, the maximal Zeno time can still be quite large and, therefore, a box is used rather than a flat end-effector, resulting in two benefits: (i) catching can be successful even if small bounces remain and (ii) the balancing challenge after establishing continuous contact simplifies. While the latter is addressed in Chapter 5, a closed-form solution of the first benefit is derived with Corollary 3.11. The numerical evaluation with results depicted in Figure 3.7 demonstrates that the range of potentially successful object states significantly increases. The convexity known from $\mathcal{X}_{\bar{Z}_{\max}}$, however, is lost.

The simulation results of both approaches, maximal Zeno time and maximal remaining rebound height, indicate that catching is more robust when the initial relative velocity is negative instead of zero. This observation is in contrast to previous work, which has typically targeted zero relative velocity without validation of this goal definition. Corollary 3.10 rigorously refines the described observation, suggesting a relative velocity maximizing the distance uncertainty that can be compensated for by a particular catching motion. The resulting suggestion (3.44) is based on a desired maximal Zeno time, acceleration bounds and an upper bound on the coefficient of restitution. If reliable success is achieved based on Corollary 3.10, one may then conceive of further reducing impact velocities with velocity matching $x_2(0) \to 0$, as considered in most previous dynamic catching approaches of a lower success rate.

## 3.5 Motion Deviating from the Ballistic Flight Path

So far, the velocity components of the object perpendicular to the distance measure $h$ have been neglected during parametrization and stability analysis. But, such parallel motions slowly begin for $t > 0$ and become non-negligible in magnitude when the end-effector must move away from the ballistic path with the start of P4. In the case of non-spherical objects, the first collision in P3 may induce significant motions perpendicular to $h$. These two major issues related to motions perpendicular to $h$ are considered here. Section 3.5.1 briefly discusses the problem of unpredictable velocity transformation when non-spherical objects collide with the unilateral constraint. Then, Section 3.5.2 addresses the effect of motions perpendicular to the ballistic flight path on the relative acceleration.

### 3.5.1 Velocity Transformation at Collisions

Rotation or velocities perpendicular to $h$ may be transformed into velocity in $h$-direction at collisions, whereas the outcome of such collisions with objects of an arbitrary shape is barely predictable. Even coefficients of restitution $\rho > 1$ in $h$-direction are likely to occur. On the other hand, frictional losses tend to be higher for polygonal, compared to spherical, objects . Therefore, calculations from the previous section cannot quantitatively predict success or failure for arbitrarily shaped objects. Nonetheless, Theorem 3.6 and Corollary 3.10 enable analysis of the qualitative effect of many variables. The object state after collisions, or at least $h$ and $\nu$, must be measurable to regain explicitness lost through impacts. Taking such measurements is a challenge, especially for small rebounds, and are thus beyond the scope of this thesis.

### 3.5.2 Acceleration Uncertainty

Rotation or velocities perpendicular to $h$ are the major source of U2, including errors and unmodeled changing tilt angles of the end-effector. The presented theorems and corollaries all rely on only the boundedness of relative acceleration $\gamma(t)$, which includes robustness against U2. These theorems explicitly do not require knowledge of a particular acceleration pattern. Therefore, the goal here is to conduct a case study of the influence of previously neglected effects on the range of occurring relative accelerations.

### 3.5.3 Numerical Example

Consider a cube-object starting above the end-effector as depicted in Figure 3.8 and Figure 3.9. Four scenarios are simulated where the end-effector motion is always planned for constant $\gamma^*(t) = 25\,\mathrm{ms}^{-2}$. Motions described in this second issue thus cause all the deviations of relative acceleration in the upper plots. The initial absolute part velocity

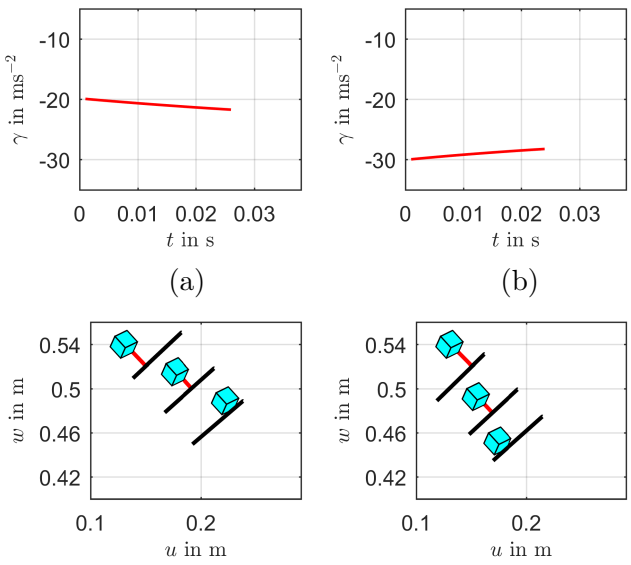


**Figure 3.8:** Simulation results to evaluate the influence on the relative acceleration in $h$-direction of motions perpendicular to $h$. The motion perpendicular to $h$ has a magnitude of $1.4\,\mathrm{ms}^{-1}$ for which it remains within the previously assumed acceleration bounds.

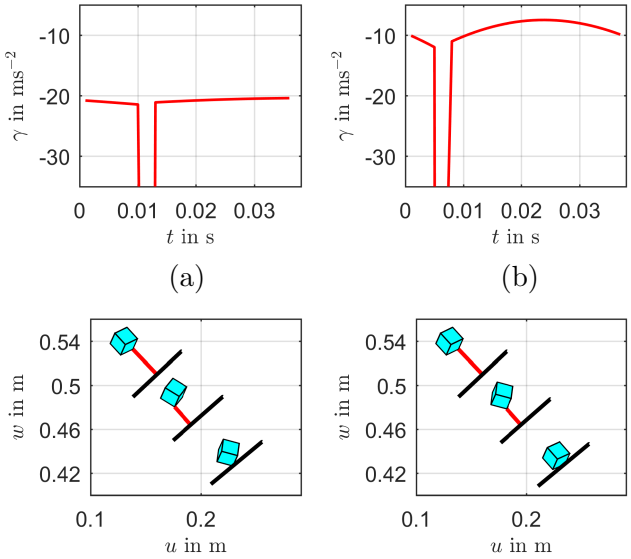


**Figure 3.9:** Simulation results to evaluate the influence on the relative acceleration in $h$-direction of object rotation. In (c), the rotation of $20\,\mathrm{rad\,s}^{-1}$ about one object axis does not violate the acceleration bounds, except for the moment when the nearest edge of the object switches. In (d), the rotation of $40\,\mathrm{rad\,s}^{-1}$ about one object axis leads to permanent violation of the acceleration bounds.

in $h$-direction is $\nu_{\mathrm{O}} = 3.5\,\mathrm{ms}^{-1}$ and the initial relative velocity is $\nu(0) = -0.3\,\mathrm{ms}^{-1}$. In the translational scenario illustrated in Figure 3.8, the relative velocity perpendicular to $h$ has a comparably large magnitude of $1.4\,\mathrm{ms}^{-1}$. As depicted in the acceleration plots, the deviation from $\gamma^*$ in this simulation is at most $\pm 5\,\mathrm{ms}^{-2}$. Hence, the conservative choice of acceleration bounds used in the example described in Section 3.4.7 covers a notably large range of unmodeled velocities perpendicular to $h$. The rotational scenarios in Figure 3.9 illustrate the effect of rotation on the relative acceleration $\gamma(t)$. Considering (2.19) and its derivatives, the closest vertex can switch, resulting in a velocity jump and thus, very large relative acceleration instants. Therefore, non-spherical parts temporarily violate (3.31) in such cases. Rapid rotations, as demonstrated in scenario (b) of Figure 3.9 and high velocities perpendicular to $h$ potentially lead to permanent violations.

## 3.6 Summary

Robotic catching represents the challenge of establishing fast contact while neither the object nor a robot end-effector have to slow down from natural or desired high operation velocities. Such a task requires manipulation planning in the presence of a complex combination of uncertainties, which may not be easily neglected. Hence, establishing contact provably robust demands a provable compensation of a non-zero set of uncertainties. At the example of robotic catching, this chapter handles the overall complexity by dividing the problem of establishing fast contact into two distinct types of problems: the relative motion between an object and end-effector on the one hand (*task level*), and the kinematic motion of the robot on the other (*joint level*). The different dynamics and constraints of both problems can thus be individually addressed with suitable methods.

When continuous contact between an object and unilateral constraint, both with state uncertainty, is to be established, as on task level during catching, a sequence of collisions during finite time inevitably applies. This is the case because the sum of all uncertainties neither allows for precise distance measurement nor for precise relative velocity measurement or estimation, which ultimately causes at least small rebounds. Building on this fundamental observation, the methods and discussions presented in this chapter contrast previous work, which has typically aimed for ideal (i.e., without any rebounds) transitions into continuous contact by velocity matching. As such, these previous works lack formulating conditions that must be fulfilled to guarantee asymptotic stability (i.e., convergence) during this non-instantaneous contact transition. The hybrid dynamical system framework briefly reviewed in Section 3.2 is well suited to formulate the stability problem for a decoupled one-dimensional bouncing ball by including the notion of Zeno behavior.

The gap between these one-dimensional considerations and the three-dimensional catching problem is addressed by dimensionality reduction and a real-time capable

dynamical system motion planner, respectively. After reduction to one dimension, the relative bouncing ball is formulated as a hybrid dynamical system with set-valued acceleration as an input. The variety of practical uncertainties reduces to a combination of three classes of uncertainties in the one-dimensional case: (U1) initial state uncertainty when the end-effector starts tracking the ballistic path, (U2) dynamic uncertainties represented by the set-valued acceleration input and (U3) uncertainty in the collision model. Quantifiable robustness against these three types of uncertainty is achieved with extensions based on uniform Zeno asymptotic stability and an associated exact tight bound on the maximal Zeno time.

A first extension demonstrates that the maximal Zeno time strictly increases as the coefficient of restitution increases, and the practical conclusion is that if catching can be guaranteed for a particular material combination of an object and end-effector, the same catching motion will also be successful in situations with more energy loss at collisions. This result is not limited to the catching example and has the potential to overcome the expensive identification of collision models in other contact situations. The second extension addresses potential users because it proposes a desirable initialization of the relative velocity such that potential distance uncertainties are maximally compensated. Moreover, this extension reveals that robustness against state uncertainty relies on small, but intentional, impacts, which requires objects to be not too fragile. The last extension formulates a bound on the maximal rebound height given the current relative state. As in practice, the end-effector might be a large box rather than a flat plate, and the maximal rebound height can be associated with the box's height to enlarge the set of potentially successful initial relative states.

The main limitation of the proposed dimensionality reduction for ballistic catching are the neglected motions perpendicular to the ballistic flight path. These deviating motions act with an offset on the relative acceleration input of the one-dimensional model and can thus cause violations of previously assumed bounds on the relative acceleration. A realistic simulation of the object and end-effector evaluates the potential offset to provide an idea of how large the acceleration set should be assumed in the one-dimensional analysis. Moreover, the simulation allows illustration of the acceleration effect of rotating polygonal objects when the object edge closest to the end-effector switches.

As this chapter has shown, contacts can be established fast and provably robust in the presence of considerable state uncertainties and without the need for precise collision modeling. The methods and results presented in this chapter may also find applications beyond the particular problem of object catching. For example, the convergence criteria for UZAS also apply in the presence of large relative velocities. Hence, an appropriate negative relative acceleration, similar to what was presented in this chapter, may enable fast and robust picking of static objects with no need to adapt or reduce the end-effector velocity before contact is established.

# 4 Joint Trajectory Planning for Manipulation through Intermittent Contacts

This chapter introduces the notion of additional, dynamically unconstrained joints. These virtual joints become available with nonprehensile dynamic manipulation via intermittent contacts. The redundancies gained with the new joints are exploited to enable faster end-effector motions of existing robots on, or close to, a known trajectory or path. Two concepts to resolve these redundancies are presented here. In the first concept, a constrained dynamic optimization resolves the redundancy using a parametrization with Hermite splines. The Hermite splines allow for an intuitive and efficient limitation of the search space and are, therefore, well suited for optimization-based offline motion planning, which also guarantees the maintenance of all joint constraints. In the second concept, inverse differential kinematic trajectory planners are compared unbiased and tuning-free based on weight optimization. From formulation until evaluation, particular attention is afforded to the infinite velocity and acceleration potential of the new virtual joint notion. A planar catching experiment presented in Chapter 5 demonstrates how the new joints enable fast, accurate and still flexible task execution with a formerly non-redundant robot.

## 4.1 Introduction and State of the Art

Repeatable success of robotic manipulation depends on robust solutions in perception, modeling and control. Robustness alone, however, is only a necessary criterion for the practical applicability of individual manipulation solutions, also referred to as *primitives*. Only in combination with efficiency (for example, throughput, range and cost) can gaps between academic and commercial interests be bridged. As an example, throwing and catching parts has the potential to increase the efficiency of industrial object transport. Nonetheless, published work regarding this problem has so far been forced to make trade-offs between robustness and efficiency when designing and conducting experiments.

Given a manipulation primitive candidate, typically formulated in Cartesian task space, experiments with redundant manipulators are carried out using one of many available inverse kinematic algorithms. Within this step from simulation to experiment, constraints on joint ranges, velocities, accelerations and torques of the manipulator joints must be met to maintain robustness claims made in task space. In the object transport example, this final step becomes challenging when a robot manipulator is

required to catch a fast flying part. Solutions to the catching problem have recently advanced from static interception of the flight trajectory [11, 46] to an adaptation of the end-effector velocity [9, 78, 85]. For provably robust catching, the manipulator end-effector might even have to follow the flight trajectory for some time with a particular acceleration pattern relative to the part. Thus far, these adapting approaches have had to reduce the end-effector velocity leading to impacts with the part [9, 78], which can cause damage. Alternatively, redundancy is resolved offline [85]. Thus, a key to further improvement is the ability to solve the inverse kinematic problem in real-time while operating close to, or even beyond, classical manipulator constraints.

The field of nonprehensile dynamic manipulation [55, 56] is well suited to address insufficient manipulator dynamics. Unlike grasping approaches [46], a generic nonprehensile end-effector provides a large potential contact area. This area can augment the kinematics by additional virtual prismatic joints for those manipulation primitives based on impacts or transitions into continuous contact. Most importantly, these virtual joints are unconstrained with respect to velocity or acceleration. After a transition into continuous contact, such as in catching, work on nonprehensile rolling manipulation [53, 75] becomes applicable. Further examples of manipulation with intermittent contacts besides catching are juggling [54, 69, 72, 81] and batting [2, 32, 86].

A body of work already exists for resolving redundancies with joint constraints, including limitations on velocity and acceleration. The most common approach is the use of a weighted pseudo-inverse that locally minimizes joint velocities in real-time. In the presence of joint displacement limits, the weights are usually formulated joint dependent [20], or the gradient of a cost function is projected into the null space of the inverse kinematic solution [51]. When a joint still moves close to a limit, the pseudo-inverse approaches can no longer guarantee accurate task execution, which can be countered by iterative joint velocity saturation [29]. The dynamic scaling of joint trajectories [40] or the task [3] is also practical to accurately track a path. If the task is known a priori, optimal control methods in combination with trajectory deformation are applicable [66, 84], which no longer allow for real-time acceleration changes.

In the field of robotic manipulation, performing particular tasks with the same agility, speed and precision as humans is still an open problem. By actively considering the nonprehensile nature as similar to the way humans approach many such challenging tasks, standard robots can be made capable of solving these tasks with the same or even better reliability. Two approaches are presented in this chapter to exploit the nonprehensile nature of tasks with intermittent contacts. The contributions are as follows:

(i) The notion of additional, dynamically unconstrained joints is introduced. These virtual joints become available with nonprehensile dynamic manipulation via intermittent contacts. Modified constraint definitions are subsequently presented to represent the unconstrained velocity and acceleration potential of the new joints.

(ii) The redundancies gained with the new joints are parametrized offline and exploited using constrained optimization to maximize task robustness while guaranteeing kinematic and dynamic feasibility.

(iii) The redundancies gained with the new joints are used to enable fast end-effector motions on, or close to, a known path with real-time acceleration input.

(iv) Inverse differential kinematic trajectory planners are compared unbiased and tuning-free based on weight optimization in the example of catching fast flying objects.

After introducing the novel notion of virtual joints, which has been published in [83], an optimization-based offline approach that integrates the task and joint level motion planning problem is presented. For this purpose, parametrizing the novel redundancy with cubic Hermite splines is proposed. A dynamic optimization with linear and nonlinear constraints then enables maximization of task goals while the resulting robot motion is guaranteed to be kinematically and dynamically feasible. The design of the optimization problem is demonstrated to be highly flexible because physical scenario parameters, such as task duration or end-effector acceleration, can easily be added to the set of optimization variables. In the challenging example of robotic catching, it is revealed that even a 2-DOF manipulator can be enabled to exactly follow a fast ballistic flight trajectory for more than 100 ms. This integrated offline redundancy resolution approach has been published in [84].

The second approach builds on standard inverse differential kinematic methods to reliably operate robots at high velocities subject to potential real-time feedback. The assumption of an approximately known reference task path or trajectory enables an automated offline optimization of design parameters, which are crucial for maintaining feasibility during operation. The proposed cost function, motivated by [100], maximizes the normalized distance to each constraint. Subject to a sufficiently large virtual joint, significant deviations from the reference path or trajectory become possible. In the example of robotic catching, two common inverse differential kinematic methods are compared unbiased as the optimization approach overcomes the need for heuristic parameter tuning. This approach has been published in [83] and was motivated by [21, 97].

## 4.2 Dynamically Unconstrained Nonprehensile Joints

For dexterous tasks such as catching objects, roboticists frequently encounter the challenging problem that motions planned for an ideal end-effector at the task level cannot be performed with the available robots [78]. The main problem is that manipulability mainly increases with the number of joints. These additional joints, i.e., motors and gears, however, add significant additional weight and inertia which must be moved by

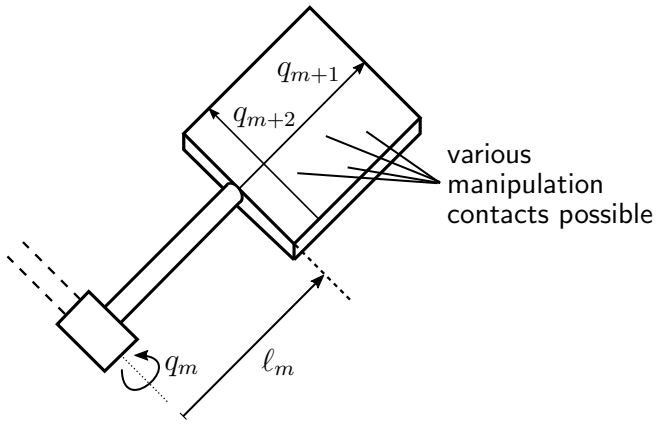


**Figure 4.1:** In nonprehensile dynamic manipulation, a generic planar end-effector (e.g. [9, 85]) at the end of the kinematic chain increases redundancy. For manipulation with intermittent contacts, such as catching or batting, these additional virtual prismatic joints provide unlimited velocity and acceleration potential.

the previous actuators in the kinematic chain. Hence, torque and power requirements also increase. As a result, solutions to challenging tasks such as catching, which require dexterous motions at high velocities, often exceed these physical limits when deployed in real robots. Success and failures during experimental validations, therefore, cannot be reliably associated with solutions at a task level.

As such, Section 4.2.1 first introduces the extension of classical kinematics with virtual nonprehensile prismatic joints for manipulation through intermittent contacts. Then, Section 4.2.2 formulates the constraints for the augmented system, which must hold to guarantee that tasks are accurately performed in experiments.

### 4.2.1 Augmented Kinematics with Unconstrained Joints

Let an $n$-dimensional manipulation task $\boldsymbol{x}_\mathrm{n} \in \mathbb{R}^n$ be solved with a manipulator that has $m$ Degrees of Freedom (DOF) denoted by $\boldsymbol{q}_\mathrm{m} \in \mathbb{R}^m$. Furthermore, assume the manipulation task requires part and end-effector to come into contact in a (temporarily) nonprehensile way. Then, there mostly exists an area, instead of a single point, on the end-effector that is eligible for the contact, see Figure 4.1. This circumstance may be reflected if $\boldsymbol{q}_\mathrm{m}$ is augmented with $a$ additional (prismatic) joints denoted by $\boldsymbol{q}_\mathrm{a} \in \mathbb{R}^a$, which leads to

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_\mathrm{m} \\ \boldsymbol{q}_\mathrm{a} \end{bmatrix}. \tag{4.1}$$

These augmented joints are typically located at the end of the kinematic chain and are limited to $a \in \{1, 2\}$.

In terms of joint constraints, the $m$ traditional joints $\boldsymbol{q}_\mathrm{m}$ and the $a$ augmented virtual joints $\boldsymbol{q}_\mathrm{a}$ reveal major differences. While not in contact with the manipulated part, the augmented joints provide – in a kinematic sense – infinite velocity and acceleration capabilities. Around the predicted contact time, velocity equality constraints may be employed to control the tangential friction effect at the collision. Alternatively, the manipulation task is robust to tangential disturbances [85]. Thus, unconstrained joint velocity applies throughout the task. The latter scenario is assumed hereafter.

### 4.2.2 Augmented Joint Constraints

Accurate tracking of a task $\boldsymbol{x}_\mathrm{n}(t)$ with a manipulator is only possible if the joint space trajectory stays within the manipulator constraints

$$\underline{\boldsymbol{Q}} \leq \boldsymbol{q} \leq \overline{\boldsymbol{Q}} \quad \text{(joint ranges)} \tag{4.2a}$$

$$\underline{\boldsymbol{V}} \leq \dot{\boldsymbol{q}}_\mathrm{m} \leq \overline{\boldsymbol{V}} \quad \text{(velocity limits)} \tag{4.2b}$$

$$\underline{\boldsymbol{T}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{T}} \quad \text{(torque limits).} \tag{4.2c}$$

Recalling the unconstrained velocity potential of the augmented joints introduced in Section 4.2.1, the velocity limits (4.2b) and torque limits (4.2c) are thus of dimension $m$, whereas the joint limits (4.2a) are of dimension $(m + a)$. The torque limits (4.2c) relate to acceleration limits using the dynamic equation

$$\boldsymbol{M}(\boldsymbol{q}_\mathrm{m})\ddot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{C}(\boldsymbol{q}_\mathrm{m}, \dot{\boldsymbol{q}}_\mathrm{m})\dot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{G}(\boldsymbol{q}_\mathrm{m}) = \boldsymbol{\tau} \ , \tag{4.3}$$

with the inertia matrix $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, the Coriolis matrix $\boldsymbol{C} \in \mathbb{R}^{m\times m}$, the gravitational vector $\boldsymbol{G} \in \mathbb{R}^m$, and the input torque $\boldsymbol{\tau} \in \mathbb{R}^m$.

The constraints (4.2) are collected in a generalized constraint vector. By defining $\boldsymbol{h} = \begin{bmatrix}\boldsymbol{q}^\mathrm{T} & \dot{\boldsymbol{q}}_\mathrm{m}^\mathrm{T} & \boldsymbol{\tau}^\mathrm{T}\end{bmatrix}^\mathrm{T}$ and therefore $\boldsymbol{h} \in \mathbb{R}^{(3m+a)}$, the constraints (4.2) become briefly

$$\underline{\boldsymbol{h}} \leq \boldsymbol{h}(t) \leq \overline{\boldsymbol{h}}, \tag{4.4}$$

where the lower and upper bounds are denoted $\underline{\boldsymbol{h}}^\mathrm{T} = \begin{bmatrix}\underline{\boldsymbol{Q}}^\mathrm{T} & \underline{\boldsymbol{V}}^\mathrm{T} & \underline{\boldsymbol{T}}^\mathrm{T}\end{bmatrix}$ and $\overline{\boldsymbol{h}}^\mathrm{T} = \begin{bmatrix}\overline{\boldsymbol{Q}}^\mathrm{T} & \overline{\boldsymbol{V}}^\mathrm{T} & \overline{\boldsymbol{T}}^\mathrm{T}\end{bmatrix}$, respectively. The constraint definition (4.4) can be extended to meet additional limitations, e.g., motor power limits.

## 4.3 Offline Integrated Motion Planning at the Task and Joint Levels

The main goal of the proposed virtual kinematic augmentation is to increase a robots degree of redundancy without changing its dynamics through the additional mass.

The new virtual joints can then be exploited to perform a main (primary) task in the presence of several secondary tasks, i.e., maintaining kinematic or dynamic constraints. For a focused evaluation of the newly proposed virtual joints, this section is limited to resolving redundancy that originates in such virtual joints. The following approach addresses this problem by first introducing a parametrization of the virtual redundancy (Section 4.3.1), which is then jointly resolved with the task level of the catching problem in an optimization program described in Section 4.3.2. A realistic numeric example in Section 4.3.3 evaluates the effectiveness of the proposed approach based on the experiments described in Chapter 5.

### 4.3.1 Redundancy Parametrization with Cubic Hermite Splines

The first concept concentrates on the redundancy originating in the novel augmented notion presented in Section 4.2.1. Hence, this section assumes that the solution to the inverse kinematic problem (2.37) of the classical $m$-DOF manipulator is known. Moreover, in the context of robust object catching and a controlled environment, as provided by a robotic throw, the task can be assumed to be sufficiently well-known for offline motion planning. Therefore, this concept approaches the redundancy resolution problem with a parametrization of the augmented joints $\boldsymbol{q}_\mathrm{a}$ to fulfill a task robustly.

The newly introduced redundancy is resolved using a normalized cubic Hermite spline

$$\begin{aligned} q_{n+a}(t, t_\mathrm{f}, \boldsymbol{p}^\mathrm{H}) = {} & \tfrac{1}{t_\mathrm{f}^3}\left(2p^\mathrm{H}_{4a-3} - 2p^\mathrm{H}_{4a-2} + t_\mathrm{f}p^\mathrm{H}_{4a-1} + t_\mathrm{f}p^\mathrm{H}_{4a}\right) t^3 \\ & - \tfrac{1}{t_\mathrm{f}^2}\left(3p^\mathrm{H}_{4a-3} - 3p^\mathrm{H}_{4a-2} + 2t_\mathrm{f}p^\mathrm{H}_{4a-1} + t_\mathrm{f}p^\mathrm{H}_{4a}\right) t^2 + p^\mathrm{H}_{4a-1}t + p^\mathrm{H}_{4a-3}, \end{aligned} \tag{4.5}$$

on the interval $t \in [0, t_\mathrm{f}]$ with 0 and $t_\mathrm{f}$ being the start and end of P3, respectively. The vector $\boldsymbol{p}^\mathrm{H} \in \mathbb{R}^{4a}$ collects the polynomial coefficients. With the Hermite spline (4.5), the virtual joints are described intuitively as the coefficients

$$\boldsymbol{p}^\mathrm{H} = \begin{bmatrix} \boldsymbol{q}_\mathrm{a}(0) & \boldsymbol{q}_\mathrm{a}(t_\mathrm{f}) & \dot{\boldsymbol{q}}_\mathrm{a}(0) & \dot{\boldsymbol{q}}_\mathrm{a}(t_\mathrm{f}) \end{bmatrix}^\mathrm{T} \tag{4.6}$$

are the displacement and the velocity of $\boldsymbol{q}_\mathrm{a}$ at start and end of P3, respectively.

### 4.3.2 Maximized Catching Task Robustness

P3 (cf. Figure 2.2) is highly crucial for nonprehensile catching, in which the robot end-effector follows the ballistic flight path of the ball. Theorem 3.5 has defined which range of the relative acceleration $\gamma(t)$ stabilizes a bouncing ball on this ballistic path. The dynamical system motion generator (3.28) with initial state (3.29) allows reconstruction of the three-dimensional end-effector motion. Not that a beneficial choice of $\boldsymbol{x}(0)$ was proposed in Section 3.4.5.

The redundancy parameters $\boldsymbol{p}^\mathrm{H}$, the final time (duration) $t_\mathrm{f}$ of P3, and the shape of $\gamma(t)$ are still undefined at this point. In the following, these remaining design parameters are chosen while being implicitly constrained by (4.2). In order to solve the offline

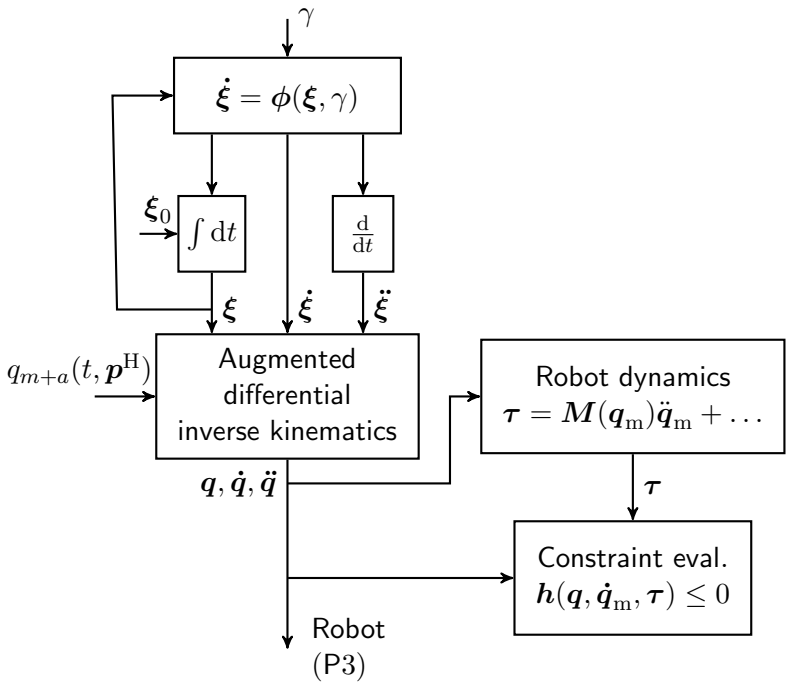


**Figure 4.2:** A block diagram visualizing the generation of the workspace motion by (3.28) based on $\gamma$ and $\boldsymbol{\xi}_0$. Using the augmented differential inverse kinematics together with $\boldsymbol{p}^{\mathrm{H}}$, the motion translates into joint space and the constraints for the dynamic optimization problem (4.7) are obtained.

motion planning problem the constrained dynamic optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}^{\mathrm{H}}, t_{\mathrm{f}}, \gamma(t)}{\text{minimize}} \quad & -\kappa \gamma_{\min} t_{\mathrm{f}} \\ \text{s.t. } & (4.3) \text{ using } (3.28) - (3.29) \text{ and } (4.5), \\ & \underline{\boldsymbol{h}} - \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \boldsymbol{\tau}) \leq \boldsymbol{0}, \ \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\mathrm{m}}, \boldsymbol{\tau}) - \overline{\boldsymbol{h}} \leq \boldsymbol{0}, \end{aligned} \tag{4.7}$$

is formulated. Additional constraints might be added depending on particular experimental setups [84]. The chosen cost function partially originates in (3.34), which quantifies the convergence speed of the Zeno behavior that inevitably occurs during any non-ideal contact transition. The proposed multiplication of the quantified convergence speed with $t_{\mathrm{f}}$ intends to maximize robustness. The most important property of this cost function is its independence from the initial relative state $\boldsymbol{x}(0)$, which accounts for the use in an uncertain environment. Nonetheless, it should be noted that a desired $\boldsymbol{x}(0)$ needs to be chosen for initializing the integration of (3.28) with (3.29) during constraint handling.

Figure 4.2 visualizes the three main parts required to evaluate the constraints of (4.7) in practice:

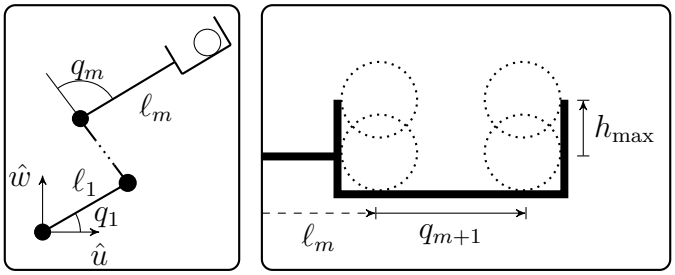


**Figure 4.3: Left:** The kinematic model for $m$ actuated rotational degrees of freedom. The $\hat{u}$-$\hat{w}$-frame coincides with the robot base. **Right:** An unactuated, box-like end-effector with height $h_{\max}$ and prismatic joint as "virtually" actuated DOF $q_{n+1}$.

(i) Generation of the workspace motion (Cartesian positions, velocities and accelerations) based on the input acceleration $\gamma$ and initial end-effector state $\boldsymbol{\xi}_0$. Here, the offline calculation allows for solving the integration problem with variable step-size solvers, which provide high and reliable accuracy. For the typically very brief duration of P3, however, the propagating errors of fixed step-size solvers are also negligible. Therefore, this part can readily be used in real-time.

(ii) Translation from the workspace into joint space motion, including redundancy resolution, for instance, using (4.5). Given a parametrization $\boldsymbol{p}^{\mathrm{H}}$ for the augmented joints, a standard inverse kinematic problem remains, which can be solved with one of the many existing methods (see [88, 90]).

(iii) Evaluation of kinematic and dynamic constraints at the joint level.

The acceleration $\gamma(t)$ must be parametrized and constraints must be evaluated at discrete time steps to solve the problem as a static optimization problem with one of the many available solvers. The particular parametrization pattern used for the input $\gamma(t)$ depends highly on the method selected to solve the problem (4.7) numerically. Typical approaches are constant, piecewise constant and piecewise linear parametrization.

### 4.3.3 Numerical Example

#### Scenario

The realistic planar scenario of a robot dynamically catching objects is regarded in its most critical phase, i.e., P3, where the robot has only $m = 2$ actuated DOFs (see Figure 4.3). Considering the given path or trajectory at the task level, as discussed in Chapter 3, this system has no redundancy, except for the choice between overarm and underarm configuration. Hence, the robotic system does not provide any degree of freedom that could be used to fulfill constraints at the joint level. However, the task of catching objects is exceedingly challenging for robots, as high velocities and

**Table 4.1:** Bounds on the optimization variables.

| Symbol | $p_1^{\mathrm{H}}$ | $p_2^{\mathrm{H}}$ | $p_3^{\mathrm{H}}$ | $p_4^{\mathrm{H}}$ | $t_\mathrm{f}$ | $\gamma$ |
|---|---|---|---|---|---|---|
| Unit | m | m | $\mathrm{ms}^{-1}$ | $\mathrm{ms}^{-1}$ | s | $\mathrm{ms}^{-2}$ |
| Upper | 0.015 | 0.015 | 2 | 2 | 0.25 | 45 |
| Lower | 0 | 0 | −2 | −2 | 0.05 | 15 |

large accelerations occur. The dynamic feasibility required to maintain robustness guarantees made at the task level can then only be achieved by task downscaling, e.g., catching from a very short range.

The virtual kinematic augmentation (4.1), here with one prismatic joint ($a = 1$) at the end of the kinematic chain, adds redundancy to the motion planning problem on joint level as the end-effector orientation can be neglected. The newly gained redundancy is then parametrized with the Hermite spline (4.5). The desired relative acceleration $\gamma$ is considered constant for $t \in [0, t_\mathrm{f}]$. The acceleration uncertainty is $\Delta\gamma = 10\,\mathrm{m\,s}^{-2}$, which defines the lower and upper acceleration bounds as $\gamma_{\min} = \gamma - \Delta\gamma$ and $\gamma_{\max} = \gamma + \Delta\gamma$, respectively. The dynamic constraints are evaluated at $N$ equally distributed discrete points in time $t_k = \frac{t_\mathrm{f} k}{N-1},\ k = 0, 1, \ldots N-1$. The release point on the throwing side is given by

$$\begin{aligned} u_\mathrm{r} = u(t_\mathrm{r}) &= -1.77\,\mathrm{m} \\ w_\mathrm{r} = w(t_\mathrm{r}) &= 0.52\,\mathrm{m} \\ \alpha_\mathrm{r} &= 37^\circ \\ \nu_\mathrm{r} &= 4.5\,\mathrm{m\,s}^{-1}, \end{aligned}$$

which complies with the experimental setup presented in Chapter 5. The optimization variables are constrained by the values in Table 4.1. The inequality constraints $\boldsymbol{h}$ from (4.4) are discretized over time with $N = 50$.

### Results and Discussion

Solving[1] (4.7) for the previously described scenario then results in $\kappa\gamma_{\min} t_\mathrm{f} = 0.12$, a constant relative acceleration $\gamma^*(t) = 39.5\,\mathrm{m\,s}^{-2}$ and P3 duration $t_\mathrm{f} = 0.101\,\mathrm{s}$. For the augmented DOF the optimization returns

$$\boldsymbol{p}^{\mathrm{H}} = \begin{bmatrix} 0.013\,\mathrm{m} & 0.003\,\mathrm{m} & -0.57\,\mathrm{m\,s}^{-1} & 0.055\,\mathrm{m\,s}^{-1} \end{bmatrix}.$$

The torques and velocities required to perform the P3 motion on the time interval $t \in [0, t_\mathrm{f}]$ are depicted in Figure 4.4. While the motion based on the augmented kinematics stays within the limits, a following motion with the standard kinematics would be dynamically infeasible. The most significant difference is observed at the

[1] Here using Sequential Quadratic Programming (SQP) in MATLAB.

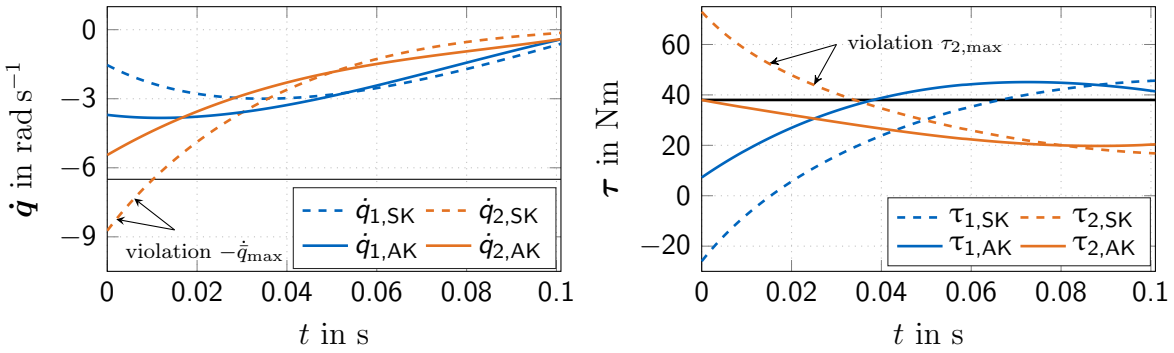


**Figure 4.4:** Required velocity and torque of the two actuated joints to perform the P3 motion. While the motion governed by the augmented kinematics (AK) stays within the limits of $\underline{\boldsymbol{V}}^{\mathrm{T}} = [-6.5 \quad -6.5]\,\mathrm{rad\,s^{-1}}$ and $\overline{\boldsymbol{T}}^{\mathrm{T}} = [54 \quad 38]\,\mathrm{Nm}$, a following motion with the standard 2-DOF kinematics (SK) would be dynamically infeasible.

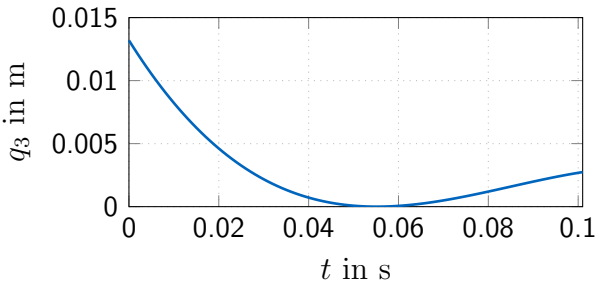


**Figure 4.5:** Resulting exploitation of nonprehensile augmented DOF $q_3$.

beginning of the following motion because here the desired velocity is the highest. In the second half, the dynamic requirements become smaller and stay within the constraints. This is owed to the fact that the desired relative acceleration results in a decreasing desired velocity. In the presented scenario, the dynamic requirements at the start of P3 are generally only met at very few points by the standard kinematics in the robot's workspace. The plot in Figure 4.5 illustrates how the AK exploit the virtual DOF to achieve dynamic feasibility. Figure 4.6 shows the maximal Zeno time based on Theorem 3.6, but here with the optimized $\gamma$ and $t_{\mathrm{f}}$. In comparison to Figure 3.6, the Zeno times with respect to the initial relative state have generally decreased.

The main drawback of the proposed optimization is the convergence of (4.7) to a feasible solution. The complex set of constraints makes it difficult to initialize the optimization program with optimization parameters that already fulfill all constraints. However, standard solvers, like the SQP method provided with MATLAB, can solve the feasibility problem jointly with the optimization problem. Moreover, considering the offline nature of the optimization-based motion planner, a large randomized set of initializations within the parameter bounds (see Table 4.1) can be evaluated efficiently

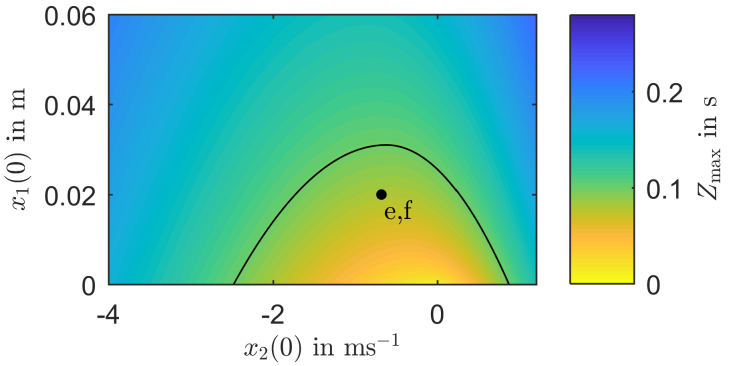


**Figure 4.6:** Contours indicating the maximal Zeno time (3.35) for acceleration bounds $\gamma(t) \in [29.5, 49.5]$ m s$^{-2}$ in P3 with the same scale as in Figure 3.6, but doubled $\Delta\gamma$ for increased robustness. The solid contour marks $Z_{\max} = 0.101$ s in view of (3.37). A further result of the optimization compared to Figure 3.6 is the reduction of the maximum occurring Zeno time from 0.28 s to 0.22 s in the depicted subspace.

with a parallelized algorithm. For the particular numeric example above, a total of 100 randomly distributed initializations converged 52 times to feasible solutions. The computation takes 414 s on a 3.9 GHz Intel Core i3-7100 with two physical cores and four threads. The number of successfully converging optimization runs scales with the requirements on task level. Amongst all task parameters, the desired initial end-effector velocity $\xi_1(0)$ from (3.29) on the ballistic path appears most critical in this context.

### 4.3.4 Discussion

This section proposes a novel, optimization-based motion planning approach that integrates the task and joint level problem of robotic catching. The main contribution of this approach is overcoming the generation of task plans that cannot be deployed in real robots because dynamic and kinematic limitations are neglected. Violations of such limitations were considered responsible for a sizable number of catching failures in the past. Successful integration originates, on the one hand, in the robust task-level approach represented by the cost function in (4.7) and, on the other hand, in the virtual joint augmentation proposed in Section 4.2.1.

The selected cost function maximizes the decrease of the Lyapunov function value of the underlying hybrid bouncing ball problem independent of particular initial relative states. As demonstrated in the numerical example, the optimization leads to a general decrease of Zeno times, which results in an increase of initial relative states that fulfill the catching condition (3.37). As a result, this approach does not require explicit modeling of the various types and sources of uncertainties and is hence, generally applicable beyond particular experimental setups.

Subject to a controlled environment that allows for approximate prior knowledge of the ballistic flight trajectory, such as using a throwing robot, the robust approach enables offline planning. Therefore, an extensive search across the parameter space can be performed to identify robust solutions at the very limit of a robot's capabilities by optimal exploitation of the newly introduced virtual prismatic joint. The numeric example in Section 4.3.3 demonstrates the significance of the virtual joint by enabling a realistic non-redundant 2-DOF robot to follow a fast trajectory through its workspace without any constraint violation. Furthermore, given this parameter search, the parametrization of the virtual joint with a cubic Hermite spline is highly efficient as it allows for a tight and intuitive limitation of the search space regarding positions and velocities.

Nonetheless, the assumption of a controlled environment and, therefore, sufficient knowledge of the ballistic flight trajectory prevents the application of this offline approach beyond a well-calibrated robot-robot scenario. Hence, larger deviations of a robotic or human throw or disturbances during flight cannot be compensated for despite the underlying task motion planner (3.28) being real-time capable. Thus, Section 4.4 proposes an alternative approach based on inverse differential kinematic planners that can operate in real-time.

## 4.4 Real-Time Motion Planning

This section compares real-time capable redundancy resolution methods to enable acceleration changes on the ballistic flight path during operation. Section 4.4.1 reviews two standard inverse differential kinematic algorithms, known to be real-time capable. The performance of these algorithms regarding constraints at the joint level strongly depends on the initial kinematic configuration and selected weighting matrices. As such, Section 4.4.2 proposes an offline optimization approach that automates the search for these weighting matrices by defining tuning goals for the algorithms. Both algorithms are compared unbiased for a realistic catching scenario in Section 4.4.3. The favored algorithm is then further analyzed in Section 4.4.4 with respect to online changing acceleration inputs.

### 4.4.1 Inverse Differential Kinematics for Redundancy Resolution

Most robot motion planning problems include a manipulator that is redundant in regard to the task its end-effector is supposed to perform. Therefore, redundancy resolution is a recurring topic for roboticists, to which Section 2.4.3 provides a brief introduction. This section reviews two standard inverse differential kinematic algorithms, known to be real-time capable. With the help of a pseudo-inverse

$$\boldsymbol{J}_{\boldsymbol{W}} = \boldsymbol{W}^{-1}\boldsymbol{J}^{\mathrm{T}}\left(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{\mathrm{T}}\right)^{-1} \tag{4.8}$$

redundancy may be resolved by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{W}} \dot{\boldsymbol{x}}_{\mathrm{n}}, \tag{4.9}$$

which locally minimizes the joint velocities according to a symmetric positive definite weighting matrix $\boldsymbol{W} \in \mathbb{R}^{(m+a)\times(m+a)}$. Additional objectives, for example respecting the limited range of the augmented joints, are included by extending (4.9) to

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{W}} \dot{\boldsymbol{x}}_{\mathrm{n}} + k \left( \boldsymbol{I}_{m+a} - \boldsymbol{J}_{\boldsymbol{W}} \boldsymbol{J} \right) \nabla w(\boldsymbol{q}), \tag{4.10}$$

which optimizes a secondary objective function $w(\boldsymbol{q})$ by projecting its gradient

$$\nabla w = \frac{\partial w(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{4.11}$$

into the null space of the Jacobian. The factor $k \in \mathbb{R}$ weights the secondary task in relation to the primary, velocity minimizing, task, whereas positive or negative signs result in maximization or minimization, respectively. Note that the joint trajectory planners (4.9) and (4.10) lead to different solutions for every possible initial configuration $\boldsymbol{q}_0$. Hence, the choice of $\boldsymbol{q}_0$ is at least as important as the tuning of $\boldsymbol{W}$ and $k$ to find feasible joint trajectories through the entire task. In practice, such tuning often turns out unsuitable and unintuitive in the presence of acceleration or torque constraints. Therefore, Section 4.4.2 formulates a constrained optimization problem to find $\boldsymbol{W}$, $k$ and $\boldsymbol{q}_0$ automatically based on a nominal task trajectory.

### 4.4.2 Maximized Distance to Constraints

Given a nominal task trajectory $\boldsymbol{x}_{\mathrm{n}}^*(t)$ with $t \in [t_0, t_{\mathrm{f}}]$, an initial joint configuration $\boldsymbol{q}_{\mathrm{m}}(t_0)$, and an initial guess for $\boldsymbol{q}_{\mathrm{a}}(t_0)$, $\boldsymbol{W}$ and $k$, the desired joint trajectory follows from numeric differentiation and integration of (4.9) or (4.10). Hence, defining a vector of optimization variables

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{a}}^{\mathrm{T}}(t_0) & \operatorname{diag}(\boldsymbol{W})^{\mathrm{T}} & k \end{bmatrix}^{\mathrm{T}}, \tag{4.12}$$

the constraint vector in (4.4) becomes $\boldsymbol{h}(\boldsymbol{x}_{\mathrm{n}}^*(t), \boldsymbol{c})$, where $\boldsymbol{W}$ is chosen a diagonal matrix. Note here that other task parameters can easily be added to $\boldsymbol{c}$, e.g. the starting point $\boldsymbol{x}(t_0)$ in the manipulator workspace.

In order to ensure that the joint trajectory not only stays within the constraints, but also keeps a distance to each constraint, an inverse distance function

$$H_i\left(h_i(\boldsymbol{x}_{\mathrm{n}}^*(t), \boldsymbol{c})\right) = \frac{\left(\bar{h}_i - h_i\right)^2}{\left(\bar{h}_i - h_i\right)\left(h_i - \underline{h}_i\right)} \tag{4.13}$$

similar to [100] is used. Note that with (4.13) the value $H_i$ becomes infinite when the $i$-th element of the vector $\boldsymbol{h}$ reaches one of its constraints. Further constraint violations

(i.e. $h_i < \underline{h}_i$ or $h_i > \overline{h}_i$) result in (4.13) becoming negative. With the help of (4.13) a dynamic optimization problem is formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{c}}{\text{minimize}} \int_{t_0}^{t_f} \sum_{i=1}^{N} H_i \left(h_i(\boldsymbol{x}_{\mathrm{n}}^*(t), \boldsymbol{c})\right) \mathrm{d}t \\ \text{s.t. } H_i \geq 0 \; \forall \; i \in \{1, 2, \ldots, N\}, t \in [t_0, t_f]. \end{aligned} \tag{4.14}$$

This automation of finding an appropriate parametrization $\boldsymbol{c}$ offline for a real-time capable pseudo-inverse kinematic planner yields the following two major benefits: (i) optimization (4.14) implicitly penalizes large velocities, which prevents singular Jacobians $\boldsymbol{J}$; and (ii) optimization (4.14) maximizes the distance to all joint constraints. Therefore, other dynamically challenging tasks $\boldsymbol{x}_{\mathrm{n}}(t)$ with $t \in [t_0, t_{\mathrm{f}}]$ can potentially be executed when close to the nominal task $\boldsymbol{x}_{\mathrm{n}}^*(t)$. Simulations described in the following section evaluate this potential flexibility.

### 4.4.3 Numerical Example: Method Comparison

Consider the translational task $\boldsymbol{x}_{\mathrm{n}} \in \mathbb{R}^2$ of tracking a ballistic flight path in the vertical plane. Therefore, the first element $x_{\mathrm{n},1}$ and the second element $x_{\mathrm{n},2}$ denote the horizontal and vertical displacement, respectively. Furthermore, as this example focuses on the joint trajectory planning, not the catching problem, it is further assume without loss of generality that object and end-effector have the same initial position $\boldsymbol{x}_{\mathrm{n,O}}(t_0) = \boldsymbol{x}_{\mathrm{n}}(t_0)$ and velocity $\dot{\boldsymbol{x}}_{\mathrm{n,O}}(t_0) = \dot{\boldsymbol{x}}_{\mathrm{n}}(t_0)$, whereas $\nu_0 := \|\dot{\boldsymbol{x}}_{\mathrm{n}}(t_0)\|_2$. Part dimensions are neglected for the same reason. Hence, the dynamical system motion planner (3.28)-(3.29) simplifies to

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} -g \sin\left(\alpha(\xi_2)\right) \\ \xi_1 \cos\left(\alpha(\xi_2)\right) \\ \xi_1 \sin\left(\alpha(\xi_2)\right) \end{bmatrix} + \begin{bmatrix} -\gamma^*(t) \\ 0 \\ 0 \end{bmatrix}, \; \boldsymbol{\xi}_0 = \begin{bmatrix} \nu_0 \\ x_{\mathrm{n},1}(0) \\ x_{\mathrm{n},2}(0) \end{bmatrix}, \tag{4.15}$$

where for now the input $\gamma^*(t) = 0$. The purpose of the input $\gamma^*(t)$ will become clear in Section 4.4.4. The dynamical system (4.15) ensures that an end-effector velocity $\xi_1$ and its respective acceleration $\dot{\xi}_1$ only act along the object's ballistic path. Solving (4.15) numerically is computationally inexpensive and thus suits well to potentially process real-time feedback, e.g. from analogue distance sensors integrated in the end-effector.

#### Scenario

Three inverse kinematic methods are compared in simulation:

S1) The non-redundant ($m = 2, a = 0$) overarm solution of a 2-DOF manipulator.

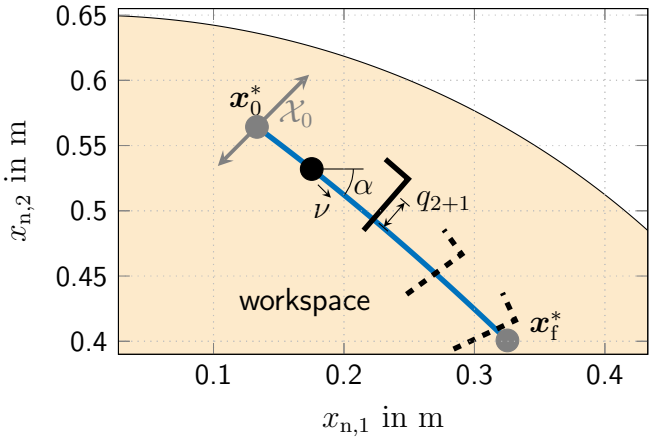


**Figure 4.7:** Segment of the part flight trajectory that must be followed by the end-effector ($\overline{Q}_3 = 1.5\,\mathrm{cm}$) for a controlled catch. The edge of the end-effector [85] compensates for post-impact motion normal to the flight path.

S2) The virtually redundant ($m = 2, a = 1$) overarm solution of a 2-DOF manipulator using the pseudo-inverse (4.8) with a diagonal weighting matrix and optimization (4.14).

S3) The virtually redundant ($m = 2, a = 1$) overarm solution of a 2-DOF manipulator using the pseudo-inverse with gradient projection (4.10), optimization (4.14), and $w(\boldsymbol{q}) = -\frac{1}{2}(q_3 - \overline{Q}_3/2)^2/(\overline{Q}_3 - \underline{Q}_3)^2$.

The three approaches are evaluated for the same task with two exploitable end-effector lengths $\overline{Q}_{3,a} = 1.5\,\mathrm{cm}$ and $\overline{Q}_{3,b} = 5\,\mathrm{cm}$, referred to as S1a, S1b, S2a, S2b, S3a, and S3b, respectively. The displacements of the joints $q_1$ and $q_2$ are considered unconstrained. The end-effector acceleration along the ballistic path is set to equal the part acceleration $\gamma^* = 0$. Here, only a short segment of $t_f = 60\,\mathrm{ms}$ ($t_0 = 0$) is considered with a large initial part velocity of $\nu_0 = 4.2\,\mathrm{ms}^{-1}$ and nominal position $\boldsymbol{x}_0^* = \begin{bmatrix}0.13\,\mathrm{m} & 0.56\,\mathrm{m}\end{bmatrix}$ in the manipulator workspace, see Figure 4.7. Together with a flight path angle $\alpha_0 = -39°$, this corresponds to a zero-height throwing point of $\boldsymbol{x}_\mathrm{n} = \begin{bmatrix}-2.15\,\mathrm{m} & 0\,\mathrm{m}\end{bmatrix}^\mathrm{T}$. All other parameters and constraints are collected in Table A.1, whereas velocity and peak torque constraints are considered symmetric.

### Results and Discussion

The left column of Figure 4.8 displays velocities and torques, which the given task requires using the well-known 2-DOF overarm inverse kinematics (S1). Horizontal lines, as in all plots that follow, highlight constraints. The length of the virtual joint is added to $\ell_2$. Obviously both velocity and torque limits are violated several times, which potentially leads to inaccurate tracking of the desired end-effector trajectory.

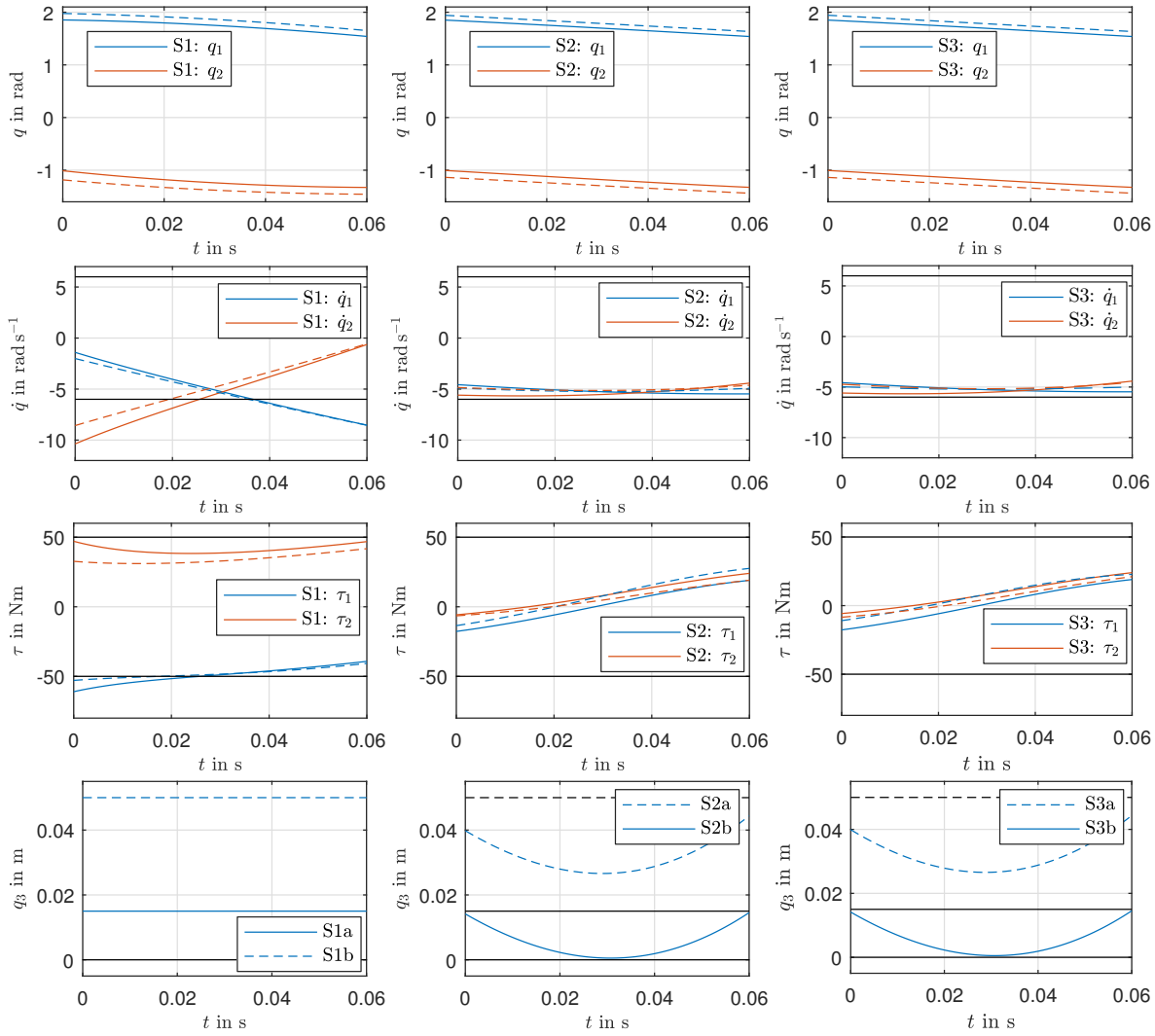


**Figure 4.8:** Simulation results for non-redundant robot (S1), redundancy resolution with weighted pseudo-inverse (S2), and with gradient projection (S3). Solid lines refer to augmentation with the shorter virtual joint $\overline{Q}_{3,a} = 1.5\,\text{cm}$ and the dashed lines to a larger joint $\overline{Q}_{3,a} = 5.0\,\text{cm}$.

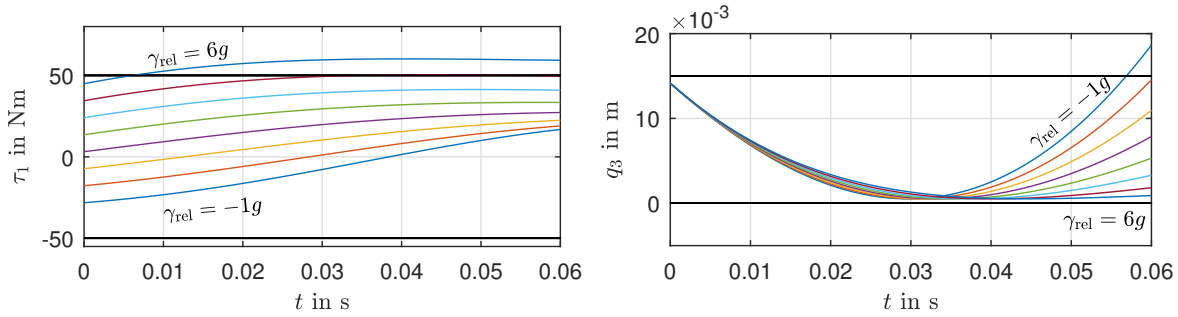


**Figure 4.9:** The effect on required joint torque and joint exploitation when applying various relative accelerations $\gamma^* \in \{-g, 0g, g, \ldots, 6g\}$ between part and end-effector.

The middle column in Figure 4.8 shows how the augmented virtual joint is exploited using (4.9) with the optimization result[2] $\boldsymbol{c}^*_{\text{S2a}} = \begin{bmatrix}0.015 & 31.0 & 21.6 & 177.7\end{bmatrix}^{\text{T}}$ and $\boldsymbol{c}^*_{\text{S2b}} = \begin{bmatrix}0.041 & 81.2 & 65.7 & 163.5\end{bmatrix}^{\text{T}}$. In comparison to S1, torque requirements are significantly reduced and velocity constraints are met. An increase from 1.5 cm to 5.0 cm in the exploitable end-effector size also shows benefits. For example, the lowest value of the joint velocity (S2a) $\dot{q}_2$ increases from $-5.67\,\text{rad}\,\text{s}^{-1}$ to $-5.14\,\text{rad}\,\text{s}^{-1}$.

The right column in Figure 4.8 shows how the augmented virtual joint is exploited using (4.10) with the optimization result $\boldsymbol{c}^*_{\text{S3a}} = \begin{bmatrix}0.015 & 31.0 & 21.6 & 177.7 & 0\end{bmatrix}$ and $\boldsymbol{c}^*_{\text{S3b}} = \begin{bmatrix}0.041 & 93.0 & 73.9 & 160.4 & 0.085\end{bmatrix}$. For the short end-effector $\overline{Q}_{3,a}$ the optimization (4.14) returns the same result ($k = 0$) as in the previous case S2a. For the longer end-effector $\overline{Q}_{3,b}$ the optimization returns a small $k = 0.085$. For example, the lowest value of the joint velocity $\dot{q}_1$ in S2b increases from $-5.26\,\text{rad}\,\text{s}^{-1}$ to $-5.22\,\text{rad}\,\text{s}^{-1}$. In both cases, convergence problems of (4.14) are observed. This is mainly due to the textbook-choice of $w(\boldsymbol{q})$, which produces large velocities in the null space to force the third joint to its range center. Such behavior is not necessarily advantageous, see also [20]. Because of the above observations, the following discussion is limited to S2a and S2b.

### 4.4.4 Numerical Example: Flexibility of Solutions

Referring to the introduction of this section, the goal is to provide joint motion planning on, or close to a given path to enable real-time acceleration input. Hence, the following question arises: Given an optimization-based pseudo-inverse solution S2a or S2b, to what extent does such a solution apply for varying accelerations $\gamma^*$? As an example, choosing $\gamma^* = g$ would cause a relative acceleration of $g$ between part and end-effector. Figure 4.9 illustrates the effect of varying $\gamma^* \in [-g, 6g]$ in steps of $g$ at the example of $q_3$ and $\tau_1$ for S2a. Using the nominal solution of $\boldsymbol{c}^*$ as above for $\gamma^* = 0$, joint trajectories

[2] using the standard SQP solver of MATLAB

stay within constraints for $\gamma^*_{,\mathrm{S2a}} \in [-0.1g, 4.9g]$ and $\gamma^*_{\mathrm{S2b}} \in [-1.1g, 5g]$. Accelerations $\gamma^* > 5g$ violate the torque limit $\overline{T}_1$ and accelerations $\gamma^* < 0.1g$ (or $-1.1g$) violate the constraint $\overline{Q}_3$. Further simulations show that all positive initial velocities $\nu_0 < 4.2\,\mathrm{ms}^{-1}$ also meet the constraints.

Manipulation systems in less structured environments may also require executing a primitive starting at several points in the manipulator workspace, e.g. catching of human throws [78]. Hence, an area around $\boldsymbol{x}^*_{\mathrm{n}}(0)$ is sought for which the solution $\boldsymbol{c}^*$ stays within all constraints. For this purpose, a simulation is performed for various initial positions on a line (Figure 4.7):

$$\boldsymbol{x}_{\mathrm{n}}(0) \in \mathcal{X}_0(x^*_{\mathrm{n}}(0), \mathcal{I}) = \left\{\boldsymbol{x} | x_2 = x^*_{\mathrm{n},2}(0) + \left(d - x^*_{\mathrm{n},1}(0)\right), d \in \mathcal{I} \subset \mathbb{R}\right\}. \tag{4.16}$$

The flexibility with respect to $\boldsymbol{x}_{\mathrm{n}}(0)$ (length of the one-dimensional set $\mathcal{I}$) highly depends on the degree of exploitation of $q_3$, whereas the influence on velocity and torque requirements is almost negligible. For $\overline{Q}_{3,a}$ the set of feasible $d$ around $x^*_1(0)$ is much smaller than 1 cm. For $\overline{Q}_{3,b}$ the feasible set becomes $\mathcal{I} = \{d | 0.12 \leq d \leq 0.21\}$ cm. Hence, one could partition a space of potential initial states and calculate an optimal parametrization for each part offline. Or, as in this particular case of planar part catching, one-dimensional manifolds for several sets $\mathcal{X}_0$. Alternatively, another offline optimization run could exploit the larger end-effector to follow the fast flight trajectory longer (larger $t_{\mathrm{f}}$) up to 85 ms instead of 60 ms.

Besides the re-computation of $\boldsymbol{c}$ in less structured environments, the proposed approach is also limited by numerical integration. In a real-time scenario, such integration uses fixed step-size solvers that cannot provide the reliable accuracy of variable step-size solvers. The same problem applies to our task planner that has to perform similar integrations when solving (4.15) in real-time. For the above described scenario with $t_{\mathrm{f}} = 60\,\mathrm{ms}$ the Euclidean workspace error remains in the negligible order of $10^{-6}\,\mathrm{m}$ using the Runge-Kutta method with a fixed step-size of 1ms. If numerical errors reach non-negligible magnitudes, closed-loop differential inverse kinematic planners (e.g. [88]) should be used instead.

Hence, the simple method (5) with offline optimized weights is already well suited for kinematic trajectory planning in the presence of dynamically unconstrained non-prehensile joints. Even with short, virtual prismatic joints, trajectories that are highly unfeasible in the classical sense may now be executed with relative accelerations on the ballistic path that change in real-time. Therefore, the optimization based pseudo-inverse method in this letter potentially enables fast and truly soft [78] ballistic catching with low impact velocities, if an appropriate task level controller for input $\gamma^*$ is found.

### 4.4.5 Discussion

Two inverse differential kinematic methods are compared in regard to exploiting the new virtual prismatic joints for ballistic catching of fast flying objects. Both methods

are real-time capable and thus, enable online changes to the end-effector acceleration on the ballistic flight path. Given both algorithm's dependence on the tuning of weighting parameters, approximate task knowledge is used in optimization. Specifically, the optimization problem automates the search for the weighting parameters such that the normalized distance to all constraints is maximized. This optimization is performed one time before operation based on a reference flight trajectory. Online deviations from the assumed reference are possible, whereas the feasible deviation extent depends on the method and distance from the constraints.

First, the effectiveness of the two inverse differential kinematic methods is verified and then compared. Recall that the null space projection method (4.10) is an extension of the standard pseudo-inverse method (4.9)-(4.8) and is typically selected to handle secondary tasks beyond the primary task of joint velocity minimization. The numeric example in Section 4.4.3 demonstrates the effectiveness of both methods, as expected. Moreover, the challenging example demonstrates that the additional design freedom of the null space method, here denoted as $k$, does not generally outperform the standard pseudo-inverse method when using the optimization (4.14) for an unbiased comparison. Of note, when a desired trajectory requires operation close to the kinematic and dynamic limits of the (virtual) joints, the null space projection generates large joint velocities, which cause joint velocity limit violation. However, the poor performance of the null space method is strongly related to the textbook selection of the secondary objective function $w(\boldsymbol{q})$.

Compared to the parametrization approach based on Hermite splines described in Section 4.3, the two main advantages of the inverse differential kinematic planners in this research are their (i) real-time capability and (ii) flexibility to cover a range of fast motions without changing design parameters. The first advantage (i) originates from the computationally inexpensive sole requirement of integrating (4.9) or (4.10) numerically. The drawback of such integration might be the propagation of numeric errors, which is irrelevant, however, for the typically short duration of ballistic catching. A numeric example in Section 4.4.4 demonstrating the integration error resulting in less than $10^{-6}\,\mathrm{m}$ Euclidean workspace error, supports this claim. In the case of non-negligible integration errors, correctional algorithms are available (see [88]). The second advantage (ii) originates from the general formulation regarding the task $\boldsymbol{x}_\mathrm{n}$. As discussed in Section 4.4.4, offline simulations reveal that a large range of accelerations on the ballistic flight path starting at high velocities can be covered if the end-effector is sufficiently large (here, 5 cm). Similar observations apply for varying starting positions $\boldsymbol{x}_\mathrm{n}(t=0)$. Requiring simulations to quantify the range of feasible variations, however, can be considered as a drawback of the inverse differential kinematic methods. Moreover, practical applications of both methods at the very limit of a robot's kinematic and dynamic capabilities have demonstrated that the Hermite spline approach can cover slightly more challenging motions (i.e., a higher velocity at the start of P3 or longer duration of P3 at the same speed). See [97] for a more comprehensive evaluation and discussion of inverse differential kinematic approaches, including the conclusion that

most singularity avoidance approaches are ultimately unsuitable for catching due to violation of the primary task, e.g., the well-known damped least-square method.

From the methods and discussion presented here, several performance improvements are possible through minor changes. The possibility to include additional scenario parameters in the parameter vector $\boldsymbol{c}$ is the most important. As an example, a robotic object throw could be represented by (parts of) the object state at the release time. An optimized reference trajectory through the workspace of the catching robot thus allows increased flexibility or catching from longer distances. Moreover, due to the flexibility discussion, the workspace of the catching robot could be separated for multiple parametrization sets. Combined with existing advanced solutions in object detection [46], flight trajectory prediction and with online capable P2/P4 trajectory planners [66], robust nonprehensile catching of human throws could become possible.

## 4.5 Summary

Teaching robots to take over tasks regularly performed by humans primarily depends on understanding how humans approach such tasks on many levels such as perception, prediction, interaction planning or joint motion planning. Most of the known approaches follow this order, which frequently results in low attention to the final step of deploying solutions to real robotic systems. Such an approach in unproblematic for many manipulation problems because the manipulation speed can simply be slowed until feasible robot motions are found, e.g., pick and place tasks. Tasks governed by naturally high operation velocities, however, fail in this case because velocity, torque or power limits are exceeded, for example, when robots are meant to catch fast flying objects. Hence, the main motivation for the methods presented in this chapter is overcoming the generation of task plans that cannot be deployed to real robots because dynamic and kinematic limitations are neglected.

Based on the observation that humans do not plan for an explicit contact point with their hand or tool during fast manipulation, a novel kinematic notion for dynamic nonprehensile object manipulation is presented. The new notion applies to manipulation via intermittent contacts and allows augmentation of traditional kinematics with up to two virtual prismatic joints. The virtual nature of the proposed joints entails additional degrees of freedom characterized by unconstrained velocity and acceleration capabilities. The only limitation that must be considered is the joint range.

Two approaches resolve the redundancy gained by the proposed virtual joints. The first approach is tailored to robust nonprehensile catching with known (differential) inverse kinematics of the traditional kinematic chain. Cubic Hermite splines parametrize the virtual, dynamically unconstrained degrees of freedom and enable intuitive and highly efficient search space limitations regarding positions and velocities. An offline optimization program then finds a parametrization that maximizes the decrease of the Lypunov function value of the underlying hybrid bouncing ball problem (recall proof

of Theorem 3.5). Additional scenario parameters such as flight trajectory shape or the robot's design can easily be added as optimization variables to increase robustness further or reduce dynamic requirements for the traditional active joints. As demonstrated with a numeric example in Section 4.3.3, this first approach leads to a general decrease of Zeno times. As a result, the set of initial relative states that fulfill the catching condition (3.37) becomes larger and thus robustness increases. Specifically, this first approach does not require explicit modeling of the various types and sources of uncertainties and is hence, generally applicable beyond particular experimental environments. Moreover, the numeric example demonstrates the significance of virtual joints by enabling a realistic non-redundant 2-DOF robot to follow a fast trajectory through its workspace without any constraint violation. Nonetheless, the assumption of a controlled environment and, therefore, sufficient knowledge of the ballistic flight trajectory prevents the application of this offline approach beyond a well-calibrated robot-robot scenario. Hence, larger deviations of a robotic or human throw or disturbances during flight cannot be compensated for despite the underlying task motion planner (3.28) being real-time capable.

To overcome some of the disadvantages of the first approach, two inverse differential kinematic methods are compared regarding the exploitation of the new virtual prismatic joints. Both methods are real-time capable and thus, enable online changes to the end-effector acceleration on the ballistic flight path. Given both algorithm's dependence on the tuning of weighting parameters, approximate task knowledge is used for searching these parameters. Specifically, an optimization problem is formulated to automate this search by maximizing the normalized distance to all constraints. This optimization is performed one time before operation based on a reference flight trajectory. Online deviations from the assumed reference are possible, whereas the feasible deviation extent depends on the method and distance from the constraints. The numeric example in Section 4.4.3 demonstrates the effectiveness of both methods as a range of fast motions can be covered without re-computing design parameters. As a future extension, partitioning the robot's workspace and computing different weighting parameters could increase flexibility. Nevertheless, requiring simulations to quantify the range of feasible variations, can be considered as a drawback of the inverse differential kinematic methods. Moreover, the challenging example reveals that the additional design freedom of the null space method not generally leads to an out-performance of the standard pseudo-inverse method when using the optimization (4.14) for an unbiased comparison. Finally, practical applications of both methods at the very limit of a robot's kinematic and dynamic capabilities have demonstrated that the Hermite spline approach can cover slightly more challenging motions.

From the methods and discussion presented in this chapter, several performance improvements are possible through minor modifications. Amongst these possibilities, including additional scenario parameters in the parameter vector $\boldsymbol{c}$ is the most important, for example, the object state at the release time. An optimized reference trajectory through the workspace of the catching robot can then allow increased flex-

ibility or catching from longer distances. Combined with existing advanced solutions in object detection [46], flight trajectory prediction and with online capable P2/P4 trajectory planners [66], robust nonprehensile catching of human throws could become possible.

# 5 Experimental Evaluation

This experimental evaluation focuses on robot-robot throwing and catching as a demonstration scenario without any visual feedback involved. Such blind robot-robot throwing and catching provide a compelling challenge to evaluate both contribution areas in this thesis, namely robust catching with explicit success bounds and robot motion planning at high speed during intermittent manipulation. Thereby, the robotic throw enables repeatedly inducing a variety of flight trajectories, which allows for testing success claims made in previous chapters. The high operation velocity during catching poses an adequate challenge for the proposed robot motion planners. Moreover, as both robots are designed to execute both tasks, i.e., throwing and catching, with the same end-effectors, test sequences longer than described in other related works become possible, allowing for statistically significant approach comparisons. Furthermore, the experiment demonstrates the potential generality of robust nonprehensile catching regarding the manageable range of object shapes.

The chapter commences with Section 5.1 to introduce the symmetric experimental setup used for the experiments. Section 5.2 briefly describes the complementary task and motion planning approaches for throwing in Section 5.2.1. Then, Section 5.2.2 states the acceleration and deceleration strategy before and after the critical tracking of the ballistic flight path. Experimental results are reported in Section 5.3, which is divided into two parts. First, the feasibility of the proposed motion planning approaches is verified. Second, multiple object shapes and different relative initial states between the object and end-effector are evaluated with a statistically significant number of task executions. Section 5.4 then summarizes the results of this chapter.

## 5.1 Symmetric Experimental Setup

Two 2-DOF robots that are symmetrically mounted in a vertical plane serve as basis for the experiments. Figure 5.1 visualizes the setup and Table A.1 summarizes the robots' kinematic and dynamic parameters that are also used for the optimizations in the previous chapters. Four simple high-gain PD-controllers robustly operate the total four robot joints at 1 kHz. By letting the joint displacement error and joint velocity error be $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$, respectively, the control law is

$$\boldsymbol{\tau} = K_{\mathrm{P}}\boldsymbol{e} + K_{\mathrm{D}}\dot{\boldsymbol{e}} \tag{5.1}$$

Throwing (Sec. 5.2.1) | Free Object Flight | Three-Phase Catching (Sec. 5.2.2)

**Figure 5.1:** Experimental setup with two symmetrical robots acting in a common vertical plane. Each robot has two actuated rotational degrees of freedom and is capable of performing both the throwing and catching task without a hardware change. The pictures illustrate, from left to right, the throwing motion, free object flight and catching motion.

with $K_\mathrm{P} = 12000$ and $K_\mathrm{D} = 100$. The joints consist of RE40 Maxon DC motors, MR Maxon (type L) 1024-bit encoders, and HFUC Harmonic Drive 1:100 gears. Hence, the joint displacement on the load side is measured with an accuracy of $1.5 \cdot 10^{-5}$ rad.

Highly simple, non-actuated, box-like end-effectors are mounted at the end of the kinematic chain, which is the major difference compared to grasping based catching. Due to this simplistic setup, the timely interception of an object's flight trajectory, with appropriate gripper orientation), is no longer sufficient for successful catching. Hence, a P3 catching motion leads to (partial) failure if the occurring uncertainties U1-U3 are not sufficiently compensated. For example, any state-of-the-art static catching approach is likely to result in a success rate of 0% with such a setup. Moreover, objects and the end-effector are likely to be damaged by the resulting collisions due to high relative velocities.

Selecting the described robot-robot scenario also enables the exclusion of human throwers and complex vision systems as potential sources for failed catching attempts. Moreover, robotic throws are repeatable, except for uncertainties, which can be assumed bounded. Given that no significant in-flight perturbations occur, experimental catching success can then be linked with critical parameters discussed in theory and simulations in previous chapters, e.g., relative acceleration or initial relative states. Offline motion planning and open-loop operation can even become sufficient for successful catching.

Due to the robustness considerations derived in this research, no visual feedback is required during operation. Nonetheless, a non-recurring calibration of the release angle $\alpha_\mathrm{r}$ and velocity $\nu_\mathrm{r}$ must be performed. The first reason for this calibration is the non-

prehensile throwing approach in which the ball does not immediately leave the box, but briefly slides along the edge of the box. During this sliding, the revolute joints perform a small angular overshoot, which depends on various parameters. However, even for a simple PD-controlled robot, the error in the release angle $\alpha_\mathrm{r}$ (here approximately 3°) does not differ notably between trials. Second, the experimental setup is not perfectly symmetrical and requires a difference in the release velocity $\nu_\mathrm{r}$ of 2% depending on the direction. This velocity is modified using standard dynamic trajectory scaling [40]. The remaining uncertainties and inaccuracies must be compensated for by a robust catch.

## 5.2 Multi-Phase Robot Motion Planning for Ballistic Catching

In P1 and P2, the throwing and catching robot accelerate from a resting position to a dynamic goal state. In P4 the catching robot decelerates from a dynamic state to a resting position. These three phases, surrounding the flight path tracking in P3, demand the solution of two-point boundary value problems of the same dynamical system (robot) with different nonlinear constraints. The method of choice here is the formulation of a constrained optimal control problem

$$\begin{aligned} \underset{\boldsymbol{\tau}}{\text{minimize}} \quad & \tfrac{1}{2}\int \boldsymbol{\tau}^\mathrm{T}\boldsymbol{\tau}\mathrm{d}t \\ \text{s.t. } & (4.3),\ \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\tau}) \leq 0,\ h^{(1,4)}(t,\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\tau}) \leq 0, \\ & \boldsymbol{h}_0(\boldsymbol{q},\dot{\boldsymbol{q}},t_0,t_\mathrm{f}) = 0 \end{aligned} \tag{5.2}$$

that penalizes large absolute torques quadratically. This approach allows to account for the appearance of large rotational velocities that require operation close to the motor velocity limits or to the peak torque limits of the gears. Indicated by the different superscripts of $\boldsymbol{h}$, P1 and P4 require task-dependent nonlinear constraints, which are described in the following.

### 5.2.1 Throwing

For throwing acceleration (P1), the goal state of the robot is determined by the desired release state of the ball. As the throw is performed without grasping, the relative acceleration in normal end-effector direction must always be negative in this phase. The calculation for this relative normal acceleration is explained in [85] and contained in $h^{(1,4)}$. The Shannon-juggler [81] end-effector design is improved from a V-shape to the shape depicted in Figure 5.2 to fix the ball in tangential direction. This design exploits centripetal forces during the rotational acceleration to fix a spherical object in a known position relative to the end-effector. Stopping at the release instant $t = t_\mathrm{r}$ is

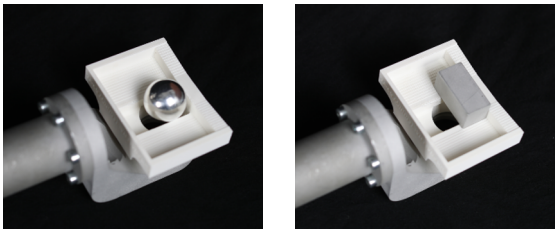

**Figure 5.2:** An end-effector design that reduces contact surface with the ball and thus avoids jamming. Spherical objects are automatically driven to the depicted throwing position, whereas cuboids must be placed manually.

realized by resetting the desired joint velocities to $\dot{\boldsymbol{q}}_{\mathrm{d}} = \mathbf{0}$. The desired joint angles are kept constant for $t \geq t_{\mathrm{r}}$, respectively. Hence, the stopping procedure of the throwing robot is entirely reliant on the high gain PD-controller reacting to the reset in the desired joint states. The resulting throwing motion is visualized in the left picture of Figure 5.1.

### 5.2.2 Three-Phase Catching

Catching is divided into the three phases (P2)-(P4), as proposed in Section 2.2.2. Of these three phases, the flight path tracking (P3) is the most crucial with regard to successful object catching. Therefore, Chapter 3 and Chapter 4 focus on P3 and the transitions from P2 and to P4. For phases P2 and P4, standard optimal control based joint motion planning (5.2) is performed offline, with linear and nonlinear constraints and motivated as follows:

(P2) For catching acceleration, the goal state of the robot is determined by the desired robot state at the beginning of the subsequent tracking motion (P3) based on the choice of the initial state in (3.28) using inverse kinematics. Unlike throwing, no normal acceleration constraint regarding the end-effector applies for this phase.

(P3) The desired end-effector motion in P3 is planned according to Chapter 3 and translated into joint motion using the methods proposed in Chapter 4.

(P4) For catching deceleration, the initial state is the last state of the path tracking motion, which does not change due to the offline P3 computation. The final state of this phase is the resting position from which the throwing motion begins accelerating in the subsequent step. During this phase, relative acceleration in the normal end-effector direction must be kept negative ($h^{(1,4)}$) to maintain stable bouncing and then continuous contact with the ball.

**Table 5.1:** Experimental Sets

| Symbol | $h(0)$ | $\nu(0)$ | Object | $h_{\max}$ | Success |
|---|---|---|---|---|---|
| Unit | m | $\mathrm{ms}^{-1}$ | - | m | - |
| $\boldsymbol{x}_{\mathrm{a}}(0)$ | 0.020 | −0.46 | Ball | 0 | 75/80 |
| $\boldsymbol{x}_{\mathrm{b}}(0)$ | 0.035 | −0.46 | Ball | 0 | 38/80 |
| $\boldsymbol{x}_{\mathrm{c}}(0)$ | 0.020 | 0 | Ball | 0 | 43/80 |
| $\boldsymbol{x}_{\mathrm{d}}(0)$ | 0.035 | −0.46 | Ball | 0.005 | 75/80 |
| $\boldsymbol{x}_{\mathrm{e}}(0)$ | 0.020 | −0.68 | Ball | 0 | **80/80** |
| $\boldsymbol{x}_{\mathrm{f}}(0)$ | 0.020 | −0.68 | Cuboid | 0 | 78/80 |

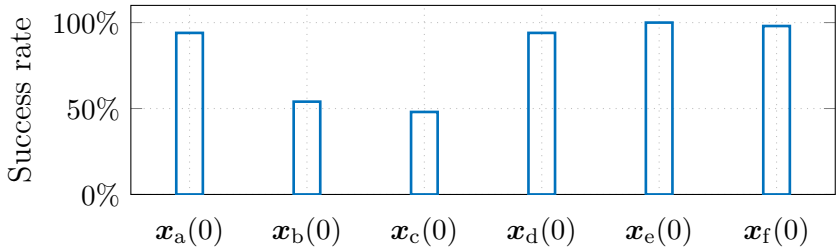


**Figure 5.3:** Successful trials out of 80 for each test set from Figure 3.6, Figure 3.7, and Figure 4.6. See Table 5.1 for details.

## 5.3 Catching Experiments

The experimental evaluation is divided into two parts. First, Section 5.3.1 confirms the dynamic feasibility of the offline motion planner and compares the robustness of different initial configurations and object shapes using offline planned throwing and catching motions. Then, Section 5.3.2 evaluates the method proposed in Section 4.4 as an alternative, which allows for real-time reactions with dynamically feasible plans for the robots' joint motions.

### 5.3.1 Catching Rigid Objects of Various Shapes

This experiment intends to validate the presented formalism concerning catching success. Moreover, this study utilizes the repeatability provided by a robot-robot experiment to evaluate the numerical robustness quantifications from Section 3.4.7 and Section 4.3.3. Thus, the six test sets $\boldsymbol{x}_{\mathrm{a-f}}(0)$ listed in Table 5.1 are performed based on the realistic simulation examples used in previous chapters. As a reference, each test set is additionally marked in Figure 3.6, Figure 3.7 and Figure 4.6. Note that calibration and throwing motion are the same for all trials and sets. The statistical results are presented in Figure 5.3 and each result is based on 80 sequential trials per set, 40 in each direction. Slow motion examples of typical catches (success and failure) are provided with the media attachment of [84].

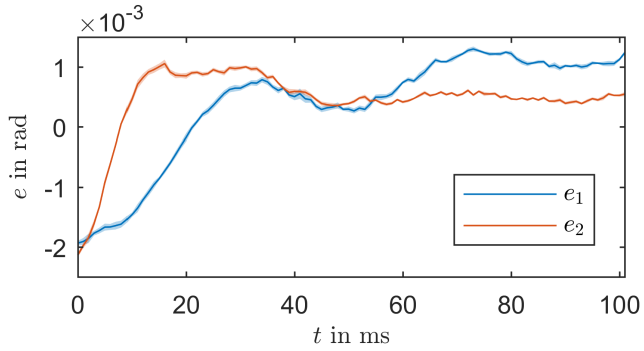


**Figure 5.4:** Tiny mean joint errors during P3 of ten $\boldsymbol{x}_{\mathrm{a}}(0)$-trials verify dynamic feasibility. The shaded areas indicate three times the standard deviation. A worst case Euclidean error in the workspace of less than 3 mm results from these errors.

The first three sets $\boldsymbol{x}_{\mathrm{a-c}}(0)$ have not undergone cost optimization (4.7) and are based on the numerical example in Section 3.4.7 using a low edge[1] box. Nevertheless, the implementation of (4.7) is used to identify a dynamically feasible joint trajectory. Figure 5.4 illustrates the small joint displacement errors. The initial relative velocity for the sets $\boldsymbol{x}_{\mathrm{a-b}}(0)$ is selected according to Corollary 3.10, whereas $\boldsymbol{x}_{\mathrm{c}}(0)$ represents velocity matching, which is a common choice in the literature. Comparing the encouraging results of $\boldsymbol{x}_{\mathrm{a}}(0)$ with the inferior results of $\boldsymbol{x}_{\mathrm{c}}(0)$, the increased robustness from using Corollary 3.10 becomes apparent. Moreover, the robustness quantification (3.35)-(3.37) correctly predicts a significantly worse performance for $\boldsymbol{x}_{\mathrm{b}}(0)$ compared to $\boldsymbol{x}_{\mathrm{a}}(0)$ because the maximal Zeno time of $\boldsymbol{x}_{\mathrm{b}}(0)$ is longer than the P3 duration. Hence, the experiment emphasizes the suitability of (3.35)-(3.37) to determine the range of potentially successful initial relative states for particular catching motions. Nonetheless, even in the controlled environment used here, various uncertainty factors exist, which prevent exact predictions of success and failure.

Test set $\boldsymbol{x}_{\mathrm{d}}(0)$ compared to set $\boldsymbol{x}_{\mathrm{b}}(0)$ validates Corollary 3.11 and, therefore, evaluates the influence of the box height on catching robustness. Noting that the initial relative state $\boldsymbol{x}_{\mathrm{d}}(0)$ fulfills the rebound height condition (3.57), but not the catching condition (3.37), the correctly predicted improvement supports the use of Corollary 3.11 to explore the effect of box height on catching success. Furthermore, the experiments validate a beneficial effect of using the cost function (4.7), which originated from the proof of Theorem 3.5.

Test set $\boldsymbol{x}_{\mathrm{e}}(0)$ evaluates the optimized P3 motion, as described in Section 4.3.3. Considering Corollary 3.10, the reduced initial relative velocity $\nu(0) = -0.68\ \mathrm{ms}^{-1}$ accounts for the changes in $\gamma$, $\Delta\gamma$ and $t_{\mathrm{f}}$. Also, the results in Table 5.1 reveal that the the optimized solution improved the already promising results from $\boldsymbol{x}_{\mathrm{a}}(0)$ to a

[1]Low edge box: edge heights of less than or equal to the distance between an object's center of mass and its farthermost point on $\mathcal{S}_{\mathrm{O}}$.

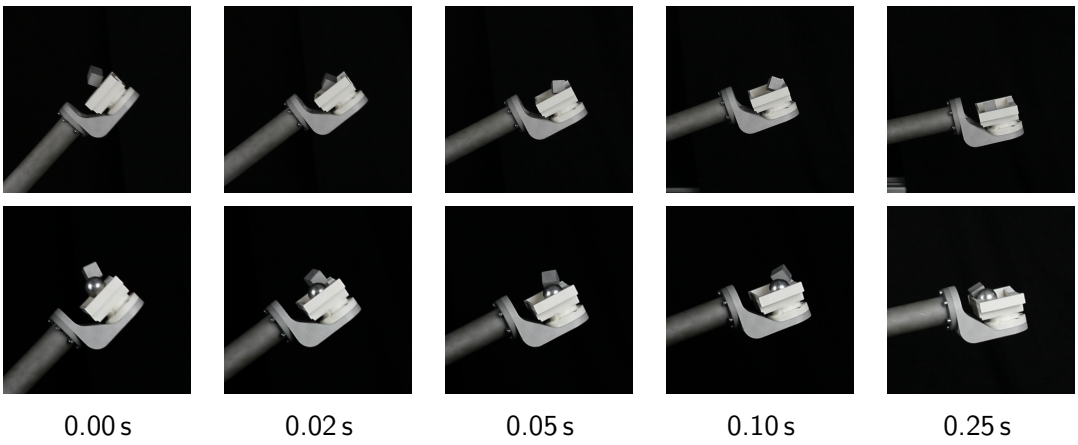


**Figure 5.5:** Frames illustrating the successful catch of a single cuboid and the successful simultaneous catch of a ball and a cube.

sequence of 80 successful catches without failure. The improvements are mainly due to the increased relative acceleration. Hence, it is inferred here that the presented approach sufficiently compensates for the occurring uncertainties U1-U3 in case of a rigid ball object.

The cuboid depicted in Figure 5.2 is used in set $\boldsymbol{x}_{\mathrm{f}}(0)$ to evaluate the potential generalizability to arbitrary object shapes. The same robot motions as in the previous set $\boldsymbol{x}_{\mathrm{e}}(0)$ are used here, which provides the most substantial robustness against the unpredictable collision effects. In contrast to the ball experiments, an operator must relocate the cuboid after every catch to the throwing position depicted in Figure 5.2 because the end-effector design automates relocation for spherical objects only. The result is 78 successful catches in a sequence of 80 using a low edge box. The frames in Figure 5.5 illustrate a successful cuboid catch. Due to these promising results, the Zeno-based approach is believed to constitute a suitable basis for future rigorous robustness claims in catching arbitrarily shaped objects.

### 5.3.2 Feasibility and Flexibility of Solutions

This section aims to verify the dynamic feasibility and flexibility claimed in Section 4.4. Offline solving of two-point boundary value problems serves as a joint motion planner to reliably move the robot from rest to the start of the tracking phase at $t_0$ and from the end $t_f$ to a resting position. The ball in Figure 5.6 is thrown in a repeatable way by another robot as illustrated in Figure 5.1, which relates to the assumption of an approximately known path.

For the fast trajectory tracking experiment, the motion planner (4.15) is implemented, where the initial states and $\boldsymbol{c}^*_{\mathrm{S2a}}$ are identical to the simulation in Section 4.4.3. The desired input of relative acceleration during tracking of the ballistic trajectory is

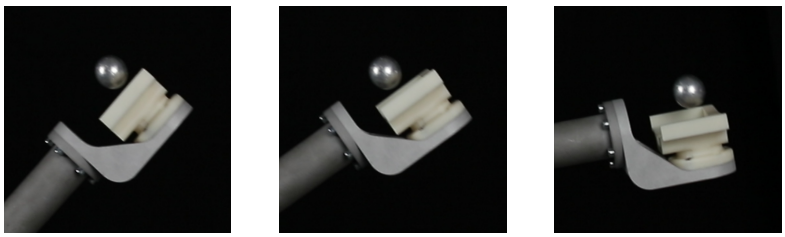

**Figure 5.6:** Exploitation of the virtual nonprehensile joint $q_{2+1}$ (c.f. Figure 4.7) of length $\overline{Q}_{3,a}$ during flight trajectory tracking illustrated by three frames taken at approximately 0 ms, 30 ms and 60 ms. Ball and end-effector have approximately the same initial velocity of $v_0 = 4.2\,\mathrm{ms}^{-1}$.

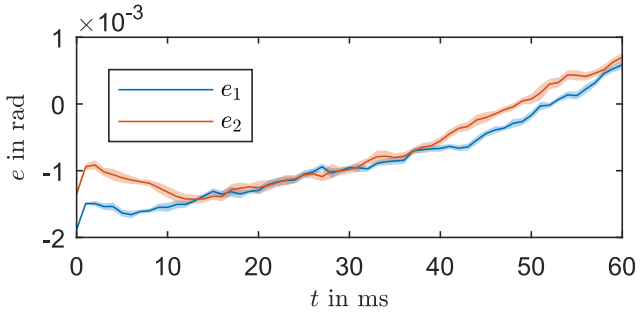


**Figure 5.7:** Mean joint errors of ten trials of flight trajectory tracking, where the shaded areas indicate twice the standard deviation. From these errors results a worst case Euclidean error in the workspace of less than 2 mm.

divided into the two parts

$$\gamma^* = \begin{cases} 0 & \text{for} \quad t \in [0, 30)\,\mathrm{ms}, \\ 4g\frac{t-30\,\mathrm{ms}}{30\,\mathrm{ms}} & \text{for} \quad t \in [30, 60]\,\mathrm{ms}, \end{cases} \tag{5.3}$$

whereas the first half was also assumed in the off-line optimization that resulted in $\boldsymbol{c}^*_{\mathrm{S2a}}$. In the second half, the ramp simulates an input varying in real-time that requires the robot to decelerate such that the relative acceleration between object and end-effector increases to $\gamma^*(t_f) = 4g$. Such variation was not considered during the off-line optimization (4.14), but lies well within the bounds discussed in Section 4.4.4.

Figure 5.6 and the multimedia attachment in [83] illustrate the resulting exploitation of the virtual nonprehensile joint $q_{m+1}$. Note that only the end-effector position ($n = 2$) and not its orientation is part of the task planner. Preliminary experiments in [85] have revealed that this approach can be sufficient for nonprehensile catching.

Based on ten trials, Figure 5.7 depicts the errors of the two actuated joints $q_1$ and $q_2$ during the tracking phase. The notably low values of these joint errors relate to workspace errors below 2 mm and thus confirm dynamically feasible trajectory planning by (4.9) and (4.14). Note that, especially from $t = 30\,\mathrm{ms}$, where the unforeseen acceleration starts, no negative effects on the tracking performance occur.

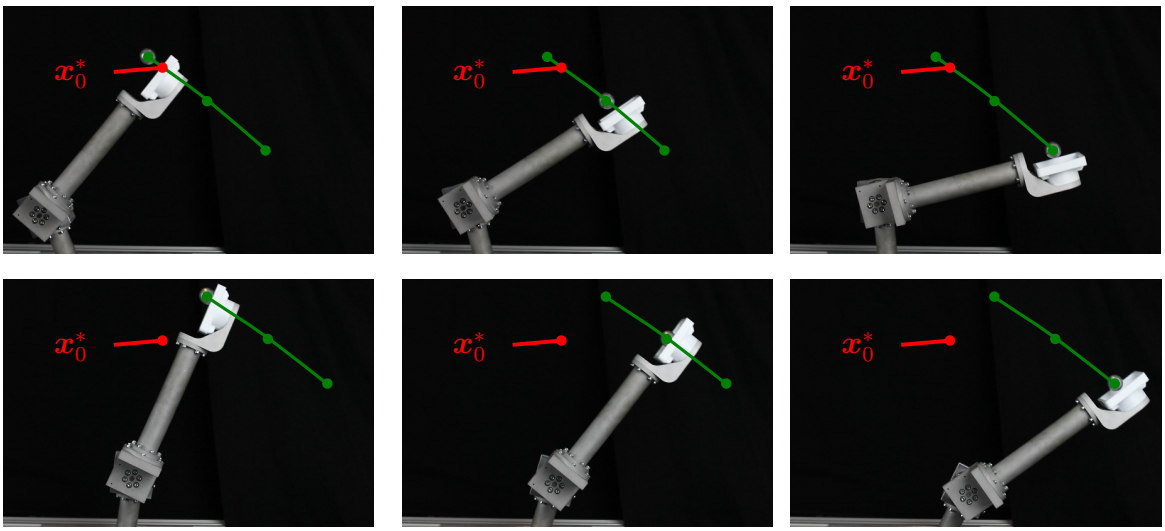


**Figure 5.8:** Tracking fast ballistic trajectories with a long ($\overline{Q}_{3,b}$) nonprehensile end-effector, acceleration $\gamma^* = 0\,\mathrm{ms}^{-2}$ for $t \in [0, 60]\,\mathrm{ms}$ and $v_0 = 4.2\,\mathrm{ms}^{-1}$. Frames are taken at approximately 0 ms, 30 ms and 60 ms. The upper row shows the reference motion ($d = 0.13\,\mathrm{cm}$), whereas the lower row reveals accurate tracking from a different starting point ($d = 0.21\,\mathrm{cm}$) while using the same $\boldsymbol{c}^*_{\mathrm{S2b}}$ (see also Figure 4.7).

Figure 5.8 illustrates the flexibility regarding the initial state. Here, a nonprehensile end-effector of length $\overline{Q}_{3,b}$ enables tracking of fast trajectories from different initial points without re-computation of $\boldsymbol{c}^*_{\mathrm{S2b}}$. As a result, future work in nonprehensile manipulation with intermittent contacts might now be able to maintain in experiments the stability and robustness claims made at the task level.

## 5.4 Summary

Taking the final development step in robotics from a simulative verification to an experimental proof of concept is frequently far from satisfactory regarding reliable task execution. Typically, a considerable gap remains between the proposed methods and their usefulness in practice. One common reason for this gap in the context of dexterous manipulation is that the kinematic and dynamic capabilities of participating robots are not considered in all planning stages. Another reason for limited experimental success in manipulation are unrewarding goal formulations that do not consider the environment state to be inherently uncertain. Therefore, the experimental results presented in this chapter intend to demonstrate that the methods proposed in previous chapters in their entirety are sufficient to solve robotic catching with non-negligible uncertainties robustly.

First, the kinematic and dynamic feasibility of the offline approach is evaluated, which provides the necessary basis for robust task execution. Here, sufficient accuracy can be confirmed by joint errors that lead to a maximal Euclidean end-effector deviation of 2 mm. In the subsequent experiments, various relative initial states are compared with statistically significant 80-trial sets. The results demonstrate that an initial negative relative velocity according to Corollary 3.10 for the tracking phase P3 increases catching robustness compared to velocity matching pursued in previous works. Moreover, Corollary 3.11 correctly predicts the increased chance of success when the height of the catching box is increased. The generality of the approach is emphasized in the example of simultaneously catching two objects with different shapes. However, the offline task and robot motion planning restricts the results to a controlled setting as typical for industrial applications.

The second evaluation addresses the real-time capable motion planner that intends to close the gap between the controlled setting in this research and existing vision-based solutions. This motion planner enables real-time acceleration changes on the known or predicted flight path at typical (high) velocities. Furthermore, the second motion planner resolves the inverse kinematic problem in a dynamically feasible way for a considerable set of flight paths and acceleration profiles.

In addition to the implementations presented in this chapter, a selection of state-of-the-art flight tracking and prediction methods could be added to catch objects thrown by humans. The real-time method derived in this research is also ready to realize closed-loop catching. Capacitive sensors or the rapidly evolving radar sensors might be able to provide necessary measurements for optimally controlling the inevitable sequence of collisions.

# 6 Conclusions and Future Work

Catching flying objects with robotic manipulators is a challenging task that demands fast and effective solutions in several robotic research fields. Therefore, robots catching an object is a frequently considered testbed to demonstrate object tracking performance combined with motion planning in highly dynamic environments. The brief time for perception and the presence of fast flight motions typically limit measurement accuracies for the shape, pose and velocities of manipulated objects. Hence, the object motion prediction after one or several measurements contains significant uncertainty. The main consequence of such uncertain dynamics in robotic manipulation are unintended collisions with other objects or the environment. Nevertheless, such collisions are inevitable, especially when continuous contacts are to be established. Each of these collisions, in turn, introduces additional uncertainty because collisions cause state changes highly sensitive to an object's shape, pose and velocity.

## 6.1 Conclusions

The main body of research related to robots catching objects, as reviewed in Chapter 2, neglects the inevitable occurrence of impacts for establishing contact at high velocities. Moreover, the proposed motion planners often overlook the limited dynamic capabilities of robotic manipulators. Each of these simplifications alone impedes formal guarantees of reliable and successful task execution, i.e., robustness. Chapter 2 to Chapter 4 address the resulting gap between theoretic research and practical applicability, as demonstrated by the experiments described in Chapter 5.

The complex problem of catching fast flying objects with a robotic manipulator is formally defined and structured in Chapter 2. The introduction of two conceptual dimensions, i.e., two physical levels and four temporal phases, efficiently decouples the robotic catching problem. The first level refers to task planning, which focuses on the relative system between a catching device (end-effector) and flying objects, while robot dynamics are mostly ignored. Introducing the shortest distance between an end-effector and object surface as a key parameter, together with its two time derivatives, has proven most generic for task level considerations. The proposed distance definition enables aggregation of the various sources of uncertainties within three representative uncertainty types. Compensating for these uncertainty types, which also include the implications of impacts, constitute a major focus of this research, as addressed in Chapter 3. The second level refers to robot motion planning, which focuses on the reliable performance of the task plan. Here, the challenge lies in providing real-time flexibility

to the task planner, while operating the robot close to its dynamic capabilities, but without ever exceeding these limits. Chapter 4 focuses on this second level. Moreover, four phases separate both levels into four time intervals, each of which has different requirements for both levels. These four phases are throwing, catching acceleration, flight trajectory tracking and deceleration.

Chapter 3 focuses on robust catching at the task level, which is related to the general problem of establishing continuous contact between an object and unilateral constraint. The state uncertainties of both, the object and unilateral constraint, inevitably cause a sequence of collisions during finite time. This is the case because the sum of all uncertainties neither allows for precise distance measurement nor for precise relative velocity measurement or estimation, which ultimately causes at least small rebounds. Building on this fundamental observation, the methods and discussions presented in Chapter 3 contrast previous work, which has typically aimed for ideal (i.e., without any rebounds) transitions into continuous contact by velocity matching. As such, these previous works lack formulating conditions that must be fulfilled to guarantee asymptotic stability (i.e., convergence) during this non-instantaneous contact transition. Building on the control theoretic hybrid dynamical system framework has proven suitable to formulate the stabilization problem for a decoupled one-dimensional bouncing ball by including the notion of Zeno behavior.

The gap between these one-dimensional considerations and the three-dimensional catching problem is addressed by dimensionality reduction and a real-time capable dynamical system motion planner, respectively. After reduction to one dimension, the relative bouncing ball is formulated as a hybrid dynamical system with set-valued acceleration as an input. As a main result of the dimensionality reduction, the variety of practical uncertainties reduces to a combination of three classes of uncertainties in the one-dimensional case: (U1) initial state uncertainty when the end-effector starts tracking the ballistic path, (U2) dynamic uncertainties represented by the set-valued acceleration input and (U3) uncertainty in the collision model. Quantifiable robustness against these three types of uncertainty is achieved with extensions based on uniform Zeno asymptotic stability and an associated exact tight bound on the maximal Zeno time. The main limitation of the proposed dimensionality reduction for ballistic catching are the neglected motions perpendicular to the ballistic flight path. These deviating motions act with an offset on the relative acceleration input of the one-dimensional model and can thus cause violations of previously assumed bounds on the relative acceleration. A realistic, three-dimensional simulation in Section 3.5 and the experiments in Chapter 5, however, indicate that typical flight motions unlikely trigger such violations.

A first extension to the UZAS notion demonstrates that the maximal Zeno time strictly increases as the coefficient of restitution increases. The practical conclusion from this result is that if establishing contact can be guaranteed for a particular material combination of an object and end-effector, the same catching motion will also be successful in situations with more energy loss at collisions. This result is not limited

to the catching example and has the potential to overcome the expensive identification of collision models in other contact situations. A second extension addresses potential users because it proposes a desirable initialization of the relative velocity such that potential distance uncertainties are maximally compensated. Moreover, this extension reveals that robustness against state uncertainty relies on small, but intentional, impacts, which requires objects to be not too fragile. A last extension formulates a bound on the maximal rebound height given the current relative state. As in practice, the end-effector might be a large box rather than a flat plate, and the maximal rebound height can be associated with the box's height to enlarge the set of potentially successful initial relative states.

Chapter 4 focuses on the dynamically feasible motion planning for a robotic manipulator to robustly perform the task level plan. A fast dynamic manipulation task, such as catching, is characterized by naturally high operation velocities, which cause standard state-of-the-art approaches to exceed velocity, torque or power limits. Hence, the main motivation for the methods presented in Chapter 4 is overcoming the generation of task plans that cannot be deployed to real robots because dynamic and kinematic limitations are too challenging or even neglected.

Based on the observation that humans do not plan for an explicit contact point with their hand or tool during fast manipulation, a novel kinematic notion for dynamic nonprehensile object manipulation is presented. This notion generally applies to manipulation via intermittent contacts and allows augmentation of traditional kinematics with up to two virtual prismatic joints. The virtual nature of the proposed joints entails additional degrees of freedom characterized by unconstrained velocity and acceleration capabilities.

Two approaches resolve the redundancy gained by the proposed virtual joints. The first approach is tailored to establishing nonprehensile contact fast and robust, as required for nonprehensile catching. Cubic Hermite splines parametrize the virtual, dynamically unconstrained degrees of freedom and enable intuitive and efficient search space limitations regarding positions and velocities. An offline optimization program then finds a parametrization that maximizes the decrease of the Lypunov function value of the underlying hybrid bouncing ball problem derived in Chapter 3. Additional scenario parameters such as flight trajectory shape or the robot's design can easily be added as optimization variables to increase robustness further or reduce dynamic requirements for the traditional active joints. This first approach is demonstrated to result in a general decrease of Zeno times. In addition, the set of initial relative states that fulfill the catching condition (3.37) becomes larger and thus robustness increases. Specifically, this first approach does not require explicit modeling of the various types and sources of uncertainties and is hence, generally applicable beyond particular experimental environments. Nonetheless, the assumption of a controlled environment and, therefore, sufficient knowledge of the ballistic flight trajectory prevents the application of this offline approach beyond a well-calibrated robot-robot scenario. Hence, larger deviations of a robotic or human throw or disturbances during flight cannot be com-

pensated for with this first approach, despite the underlying task motion planner (3.28) being real-time capable.

To overcome some of the disadvantages of the first approach, two inverse differential kinematic methods are compared regarding the exploitation of the new virtual prismatic joints. Both methods are real-time capable and thus, enable online changes to the end-effector acceleration on the ballistic flight path. Given both algorithm's dependence on the tuning of weighting parameters, approximate task knowledge is used for searching these parameters. Specifically, an optimization problem is formulated to automate this search by maximizing the normalized distance to all constraints. This optimization is performed one time before operation based on a reference flight trajectory. Online deviations from the assumed reference are possible, whereas the feasible deviation extent depends on the method and distance from the constraints. A numeric example demonstrates the effectiveness of both methods as a range of fast motions can be covered without re-computing weighting parameters. Nevertheless, requiring simulations to quantify the range of feasible variations, can be considered as a drawback of the inverse differential kinematic methods. Moreover, the challenging example reveals that the additional design freedom of the null space method not generally leads to an out-performance of the standard pseudo-inverse method when using automated weight optimization for an unbiased comparison.

Chapter 5 presents an experimental validation of robust catching using the methods proposed in previous chapters. First, the kinematic and dynamic feasibility of the offline approach is approved. In the subsequent step, various relative initial states are compared in nonprehensile robot-robot throwing and catching with statistically significant 80-trial sets. This experiment contrast previous work because the robot throw enables repeatable testing for specific degrees of uncertainty. As such, the explicit robustness bounds, previously claimed in theory and simulation, can also be validated in practice. Findings demonstrate that an initial negative relative velocity during the transition to tracking phase P3 increases catching robustness compared to velocity matching as pursued in previous works. The generality of the approach is emphasized in the example of simultaneously catching two objects with different shapes. The second evaluation demonstrates that the real-time motion planner enables end-effector operations on the known or predicted flight path at typical high velocities. Furthermore, the real-time motion planner resolves the inverse kinematic problem in a dynamically feasible way for a considerable set of flight paths and acceleration profiles.

## 6.2 Directions for Future Research

The contributions provided by this research make it possible to robustly establish contacts in dynamic environments with high operation velocities within controlled settings. Thus, pursuing the following research directions is recommended to alleviate restric-

tions for robust catching and beyond this, for dynamic manipulation environments in general:

- **Improve catching robustness:** Thus far, provable robustness in this research is limited to catching spherical objects, whereas the provided experiments indicate that generalizations to arbitrary object shapes do exist. Moreover, the methods applied in this research readily enable closing the control-loop for catching at task level. This motivates further research to minimize the Zeno times of the hybrid bouncing ball system with an uncertain but controllable relative gravitational field as input. Achieving performant real-time sensing of the relative motion constitutes a strong enabler in this context. Here, one may profit from the currently rapid progress in radar sensor development, which is mainly driven by the demands of the automotive industry. Alternatively, the fast measurement from analogue capacitive sensors can be used at short-range.

- **Set-valued impact modeling:** The classic impact modeling literature focuses on the accurate prediction of post-impact states regarding the pre-impact situation. Motivated by the set-valued bouncing ball models used in this research and the cited references, the field of dynamic manipulation is likely to benefit from developing set-valued impact models based on well-defined uncertainty in the pre-impact state measurements.

- **Kinodynamically optimized manipulators for dynamic manipulation:** A typical characteristic of dynamic manipulation is the requirement for large accelerations at the end-effector, which are often paired with fast motions, such as in the catching scenario used in this research. Thus, the kinematic and dynamic design of manipulators should be rethought to avoid constraint violations. Extending the set of optimization variables with kinematic parameters, such as link lengths, and including the related changes of the dynamic model seems a promising way to increase the action space of manipulators for dynamic manipulation tasks, for example, to increase the throwing range.

- **Extension towards fast picking:** The problem of catching fast flying objects may be considered a general challenge to establishing fast and robust contacts. Thus, the methods applied in this research, with suitable extensions, might show useful for quickly grasping resting objects. Most relevant here are the stability conditions of the relative system between an object and end-effector, and the exploitation of the free choice of contact as part of the robot motion planning.

# A Appendix

## A.1 Parameters and Constraints of the Experimental Setup

**Table A.1:** Parameter and constraint values for 2-DOF

| Symbol | Value | Quantity |
|---|---|---|
| $\ell_1$ | $0.315\,\mathrm{m}$ | Length of 1st link |
| $\ell_2$ | $0.320\,\mathrm{m}$ | Length of 2nd link |
| $\rho$ | 0.35 | Coefficient of restitution |
| $r$ | $0.015\,\mathrm{m}$ | Ball radius |
| $g$ | $9.81\,\mathrm{m\,s^{-2}}$ | Gravitational constant |
| $\Delta\gamma$ | $10\,\mathrm{m\,s^{-2}}$ | Acceleration range |
| $\alpha_\mathrm{r}$ | $37°$ | Release angle |
| $t_\mathrm{r}$ | $-0.534\,\mathrm{s}$ | Release time |
| $u_\mathrm{r}$ | $-1.770\,\mathrm{m}$ | Hor. release position |
| $w_\mathrm{r}$ | $0.525\,\mathrm{m}$ | Vert. release position |
| $\nu_\mathrm{r}$ | $4.50\,\mathrm{m\,s^{-2}}$ | Release velocity |
| $x_1(0)$ | $0.02\,\mathrm{m}$ | Initial relative height |
| $x_2(0)$ | $-0.68\,\mathrm{m\,s^{-1}}$ | Initial relative velocity |
| $\tau_{1,\max}$ | $54\,\mathrm{Nm}$ | Peak torque limit joint 1 |
| $\tau_{2,\max}$ | $38\,\mathrm{Nm}$ | Peak torque limit joint 2 |
| $\dot{\tilde{q}}_\mathrm{min}$ | $0.4\,\mathrm{rad\,s^{-1}}$ | Min. rot. speed at transition |
| $\dot{\tilde{q}}_\mathrm{max}$ | $6.5\,\mathrm{rad\,s^{-1}}$ | Max. rot. speed |
| $P$ | $120\,\mathrm{W}$ | Max. motor power |
| $q_\mathrm{min}$ | $0°$ | Box open in flight dir. |
| $q_\mathrm{max}$ | $90°$ | Box open in flight dir. |
| $q_{3,\max}$ | $0.015\,\mathrm{m}$ | Tangential box size |
| $h_\mathrm{max}$ | $0.005\,\mathrm{m}$ | Normal box size (height) |
| $N$ | 50 | Constraint evaluations |

# Bibliography


[1] Aaron D. Ames, Paulo Tabuada, and Shankar Sastry. On the stability of zeno equilibria. In João P. Hespanha and Ashish Tiwari, editors, *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC*, pages 34–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] Russell L. Andersson. Aggressive trajectory generator for a robot ping-pong player. *IEEE Control Systems Magazine*, 9(2):15–21, 1989.

[3] Gianluca Antonelli, Stefano Chiaverini, and Giuseppe Fusco. Kinematic control of redundant manipulators with on-line end-effector path tracking capability under velocity and acceleration constraints. *IFAC Proceedings Volumes*, 33(27):183–188, 2000.

[4] Brenna D. Argall and Aude G. Billard. A survey of tactile human–robot interactions. *Robotics and Autonomous Systems*, 58(10):1159–1176, 2010.

[5] Garth H. Ballantyne and Fred Moll. The da vinci telerobotic surgical system: The virtual operative field and telepresence surgery. *Surgical Clinics of North America*, 83(6):1293–1304, 2003.

[6] Klaus-Jürgen Bathe. *Finite element procedures*. Bathe, Boston, Mass., 2006.

[7] Klaus-Jürgen Bathe and Pavel A. Bouzinov. On the constraint function method for contact problems. *Computers & Structures*, 64(5-6):1069–1085, 1997.

[8] Georg Bätz. *Planning and Control Methods for Robotic Manipulation Tasks with Non-Negligible Dynamics*. Dissertation, Technische Universität München, München, 2011.

[9] Georg Bätz, Arhan Yaqub, Hayan Wu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss. Dynamic manipulation: Nonprehensile ball catching. In *Automation MED*, pages 365–370. 2010.

[10] Berthold Bäuml and Gerd Hirzinger. When hard realtime matters: Software for complex mechatronic systems. *Robotics and Autonomous Systems*, 56(1):5–13, 2008.

[11] Berthold Bäuml, Thomas Wimböck, and Gerd Hirzinger. Kinematically optimal catching a flying ball with a hand-arm-system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2592–2599. 2010.

[12] Biemond, J. J. Benjamin, Nathan van de Wouw, Heemels, W. P. Maurice H., and Hendrik Nijmeijer. Tracking control for hybrid systems with state-triggered jumps. *IEEE Transactions on Automatic Control*, 58(4):876–890, 2013.

[13] Raymond M. Brach. *Mechanical impact dynamics: Rigid body collisions*. Raymond M. Brach, rev. ed. edition, 2007.

[14] Michael S. Branicky, Vivek S. Borkar, and Sanjoy K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, 1998.

[15] Bernard Brogliato. *Nonsmooth mechanics: Models, dynamics, and control*. Communications and control engineering. Springer, London and New York, 2nd ed. edition, 1999.

[16] Mireille Broucke and Ari Arapostathis. Continuous selections of trajectories of hybrid systems. *Systems & Control Letters*, 47(2):149–157, 2002.

[17] Martin Buehler, Daniel E. Koditschek, and P. J. Kindlmann. Planning and control of robotic juggling and catching tasks. *The International Journal of Robotics Research*, 13(2):101–118, 1994.

[18] Martin Buss. *Methoden zur Regelung hybrider dynamischer Systeme: Modelle, Regelkreise, Optimalsteuerung, Rechnerwerkzeuge und Mechatronik-Anwendungen*, volume Nr. 970 of *Fortschritt-Berichte / VDI Mess-, Steuerungs- und Regelungstechnik*. VDI-Verl., Düsseldorf, 2002.

[19] M. Kanat Camlibel and J. M. Schumacher. On the zeno behavior of linear complementarity systems. In *IEEE Conference on Decision and Control*, pages 346–351. 2001.

[20] Tan Fung Chan and Rajiv V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292, 1995.

[21] Christian Czempinski. *Nonprehensile Catching: A Virtual Joint to Maximize the Catching Distance*. Bachelor's thesis, Technische Universität München, München, 2015.

[22] Pierluigi Cigliano, Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. Robotic ball catching with an eye-in-hand single-camera system. *IEEE Transactions on Control Systems Technology*, 23(5):1657–1671, 2015.

[23] Murilo G. Coutinho. *Dynamic Simulations of Multibody Systems*. Springer New York, New York, NY and s.l., 2001.

[24] Elizabeth A. Croft, Robert G. Fenton, and Beno Benhabib. Optimal rendezvous-point selection for robotic interception of moving objects. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 28(2):192–204, 1998.

[25] Agostino de Santis, Bruno Siciliano, Alessandro de Luca, and Antonio Bicchi. An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3):253–270, 2008.

[26] Koichiro Deguchi, Hironari Sakurai, and Shun Ushida. A goal oriented just-in-time visual servoing for ball catching robot arm. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3034–3039.

[27] Philine Donner and Martin Buss. Cooperative swinging of complex pendulum-like objects: Experimental evaluation. *IEEE Transactions on Robotics*, 32(3):744–753, 2016.

[28] Geir E. Dullerud and Sanjay Lall. Asynchronous hybrid systems with jumps – analysis and synthesis methods. *Systems & Control Letters*, 37(2):61–69, 1999.

[29] Fabrizio Flacco, Alessandro de Luca, and Oussama Khatib. Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Transactions on Robotics*, 31(3):637–654, 2015.

[30] Cinzia Freschi, Vincenzo Ferrari, F. Melfi, Mauro Ferrari, Franco Mosca, and Alfred Cuschieri. Technical review of the da vinci surgical telemanipulator. *The International Journal of Medical Robotics + Computer Assisted Surgery : MRCAS*, 9(4):396–406, 2013.

[31] Udo Frese, Berthold Bäuml, Steffen Haidacher, Günter Schreiber, Ingo Schaefer, Matthias Hähnle, and Gerd Hirzinger. Off-the-shelf vision for a robotic ball catcher. In *RSJ/IEEE International Conference on Intelligent Robots and Systems*, pages 1623–1629, 2001.

[32] Matthew Gardner, Yan-Bin Jia, and Huan Lin. Batting flying objects to the target in 2D. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3225–3232, Piscataway, NJ, 2016. IEEE.

[33] G. Gilardi and Inna Sharf. Literature survey of contact dynamics modelling. *Mechanism and Machine Theory*, 37(10):1213–1239, 2002.

[34] Rafal Goebel, Ricardo G. Sanfelice, and Andrew R. Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.

[35] Rafal Goebel and Andrew R. Teel. Lyapunov characterization of zeno behavior in hybrid systems. In *IEEE Conference on Decision and Control*, pages 2752–2757, 2008.

[36] Werner Goldsmith. *Impact: The theory and physical behaviour of colliding solids*. Dover Publications, Mineola, NY, originally publ. in 1960 edition, 2015.

[37] Thomas F. Heimsch and Remco I. Leine. Lyapunov stability theory for non-smooth non-autonomous mechanical systems applied to the bouncing ball problem. In *Volume 4: 7th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B and C*, pages 465–473. ASME, 2009.

[38] Michael Heymann, Feng Lin, George Meyer, and Stefan Resmerita. Analysis of zeno behaviors in a class of hybrid systems. *IEEE Transactions on Automatic Control*, 50(3):376–383, 2005.

[39] Mitsuru Higashimori, Keisuke Utsumi, and Omoto, Yasutaka Kaneko, Makoto. Dynamic manipulation inspired by the handling of a pizza peel. *IEEE Transactions on Robotics*, 25(4):829–838, 2009.

[40] John M. Hollerbach. Dynamic scaling of manipulator trajectories. *Journal of Dynamic Systems, Measurement, and Control*, 106(1):102, 1984.

[41] Won Hong and Jean-Jacques E. Slotine. Experiments in hand-eye coordination using active vision. In Oussama Khatib and J. Kenneth Salisbury, editors, *Exp. Robotics IV*, volume 223 of *Lecture Notes in Control and Information Sciences*, pages 130–139. Springer-Verlag, London, 1997.

[42] B. Hove and Jean-Jacques E. Slotine. Experiments in robotic catching. In *IEEE American Control Conference*, pages 380–386. 1991.

[43] Yan-Bin Jia. Three-dimensional impact: energy-based modeling of tangential compliance. *The International Journal of Robotics Research*, 32(1):56–83, 2013.

[44] Charles Kemp, Aaron Edsinger, and Eduardo Torres-Jara. Challenges for robot manipulation in human environments. *IEEE Robotics & Automation Magazine*, 14(1):20–29, 2007.

[45] Oussama 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.

[46] Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.

[47] Jens Kober, Matthew Glisson, and Michael Mistry. Playing catch and juggling with a humanoid robot. In *IEEE-RAS International Conference on Humanoid Robots*, pages 875–881. 2012.

[48] Roberto Lampariello, Duy Nguyen-Tuong, Claudio Castellini, Gerd Hirzinger, and Jan Peters. Trajectory planning for optimal robot catching in real-time. In *IEEE International Conference on Robotics and Automation*, pages 3719–3726, 2011.

[49] Andrew Lamperski and Aaron D. Ames. On the existence of zeno behavior in hybrid systems with non-isolated zeno equilibria. In *IEEE Conference on Decision and Control*, pages 2776–2781. 2008.

[50] Daniel Liberzon and A. Stephen Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.

[51] Alain Liégeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977.

[52] Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. 3d monocular robotic ball catching. *Robotics and Autonomous Systems*, 61(12):1615–1625, 2013.

[53] Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. The effect of shapes in input-state linearization for stabilization of nonprehensile planar rolling dynamic manipulation. *IEEE Robotics and Automation Letters*, 1(1):492–499, 2016.

[54] Kevin M. Lynch and Craig K. Black. Recurrence, controllability, and stabilization of juggling. *IEEE Transactions on Robotics and Automation*, 17(2):113–124, 2001.

[55] Kevin M. Lynch and Matthew T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999.

[56] Kevin M. Lynch and Todd D. Murphey. Control of nonprehensile manipulation. In Antonio Bicchi, H. I. Christensen, and Domenico Prattichizzo, editors, *Control problems in robotics*, volume 4 of *Springer Tracts in Advanced Robotics*, pages 39–57. Springer, Berlin and New York, 2003.

[57] Kevin M. Lynch, Michael Northrop, and Peng Pan. Stable limit sets in a dynamic parts feeder. *IEEE Transactions on Robotics and Automation*, 18(4):608–615, 2002.

[58] Zoltan-Csaba Marton, Dejan Pangercic, Nico Blodow, and Michael Beetz. Combined 2d–3d categorization and classification for multimodal perception systems. *The International Journal of Robotics Research*, 30(11):1378–1402, 2011.

[59] Matthew T. Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):1–28, 2018.

[60] Matthew T. Mason and Kevin M. Lynch. Dynamic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 152–159. 1993.

[61] D. Miller. NASA technology roadmaps: TA4: Robotics and autonomous systems, 2015.

[62] Oscar E. Montano, Yury Orlov, and Yannick Aoustin. Nonlinear $H_\infty$-control under unilateral constraints. *International Journal of Control*, 89(12):2549–2571, 2016.

[63] Yizhar Or and Aaron D. Ames. Stability and completion of zeno equilibria in lagrangian hybrid systems. *IEEE Transactions on Automatic Control*, 56(6):1322–1336, 2011.

[64] Yizhar Or and Andrew R. Teel. Zeno stability of the set-valued bouncing ball. *IEEE Transactions on Automatic Control*, 56(2):447–452, 2011.

[65] Benjamin Passenberg. *Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems*. Dissertation, Technische Universität München, München, 2012.

[66] Alexander Pekarovskiy, Thomas Nierhoff, Sandra Hirche, and Martin Buss. Dynamically consistent online adaptation of fast motions for robotic manipulators. *IEEE Transactions on Robotics*, pages 1–17, 2017.

[67] Alexander Pekarovskiy, Ferdinand Stockmann, Masafumi Okada, and Martin Buss. Hierarchical robustness approach for nonprehensile catching of rigid objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3649–3654, 2014.

[68] Luka Peternel, Nikos Tsagarakis, and Arash Ajoudani. A human-robot comanipulation approach based on human sensorimotor information. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(7):811–822, 2017.

[69] Philipp Reist and Raffaello D'Andrea. Design and analysis of a blind juggling robot. *IEEE Transactions on Robotics*, 28(6):1228–1243, 2012.

[70] Alfred A. Rizzi and Daniel E. Koditschek. Further progress in robot juggling: the spatial two-juggle. In *IEEE International Conference on Robotics and Automation*, pages 919–924, 1993.

[71] Renaud Ronsse, Philippe Lefevre, and Rodolphe Sepulchre. Sensorless stabilization of bounce juggling. *IEEE Transactions on Robotics*, 22(1):147–159, 2006.

[72] Renaud Ronsse, Philippe Lefevre, and Rodolphe Sepulchre. Rhythmic feedback control of a blind planar juggler. *IEEE Transactions on Robotics*, 23(4):790–802, 2007.

[73] J. E. Routh. *Dynamics of a system of rigid bodies*. Dover, New York, 1960.

[74] Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

[75] Ji-Chul Ryu, Fabio Ruggiero, and Kevin M. Lynch. Control of nonprehensile rolling manipulation: Balancing a disk on a disk. *IEEE Transactions on Robotics*, 29(5):1152–1161, 2013.

[76] Alessandro Saccon, Nathan van de Wouw, and Henk Nijmeijer. Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking. In *IEEE Conference on Decision and Control*, pages 3065–3070, 2014.

[77] Joe Sachs and Aristoteles. *Aristotle's physics: A guided study*. Masterworks of discovery. Rutgers University Press, New Brunswick, N.J, 2010.

[78] Seyed Sina Mirrazavi Salehian, Mahdi Khoramshahi, and Aude Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32(2):1–10, 2016.

[79] Ricardo G. Sanfelice, Andrew R. Teel, and Rodolphe Sepulchre. A hybrid systems approach to trajectory tracking control for juggling systems. In *IEEE Conference on Decision and Control*, pages 5282–5287, 2007.

[80] Aykut C. Satici, Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. A coordinate-free framework for robotic pizza tossing and catching. In *IEEE International Conference on Robotics and Automation, Stockholm, Sweden, May 16th-21st*, pages 3932–3939, 2016.

[81] Stefan Schaal and Christopher G. Atkeson. Open loop stable control strategies for robot juggling. In *IEEE International Conference on Robotics and Automation*, pages 913–918. 1993.

[82] Stefan Schaal, Christopher G. Atkeson, and Dagmar Sternad. One-handed juggling: A dynamical approach to a rhythmic movement task. *Journal of Motor Behavior*, 28(2):165–183, 1996.

[83] Markus M. Schill and Martin Buss. Kinematic trajectory planning for dynamically unconstrained nonprehensile joints. *IEEE Robotics and Automation Letters*, 3(2):728–734, 2018.

[84] Markus M. Schill and Martin Buss. Robust ballistic catching: A hybrid system stabilization problem. *IEEE Transactions on Robotics*, 34(6), 2018.

[85] Markus M. Schill, Felix Gruber, and Martin Buss. Quasi-direct nonprehensile catching with uncertain object states. In *IEEE International Conference on Robotics and Automation*, pages 2468–2474. 2015.

[86] Taku Senoo, Akio Namiki, and Masatoshi Ishikawa. Ball control in high-speed batting motion using hybrid trajectory generator. In *IEEE International Conference on Robotics and Automation*, pages 1762–1767, 2006.

[87] M. Shahid Shaikh and Peter E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Transactions on Automatic Control*, 52(9):1587–1603, 2007.

[88] Bruno Siciliano. *Robotics: Modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, London, 2009.

[89] Simplicius and David Konstan. *On Aristotle Physics 6*. Ancient commentators on Aristotle. Bloomsbury, London, 2013.

[90] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot modeling and control*. John Wiley & Sons, Hoboken, NJ, 2006.

[91] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.

[92] Tokunori Tabata and Yasumichi Aiyama. Passing manipulation by 1 degree-of-freedom manipulator - catching manipulation of tossed object without impact. In *Proceedings of the IEEE International Symposium onAssembly and Task Planning, 2003*, pages 181–186. IEEE, 2003.

[93] Naoki Uchiyama, Shigenori Sano, and Kazuaki Ryuman. Adaptive control of a robotic manipulator for soft catching of a falling object. *Mechanical Engineering Journal*, 2(3):14–00074–14–00074, 2015.

[94] Tyrone L. Vincent. Controlling a ball to bounce at a fixed height. In *IEEE American Control Conference*, pages 842–846, 1995.

[95] Gregory C. Walsh, Hong Ye, and Linda G. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10(3):438–446, 2002.

[96] Yu Wang and Matthew T. Mason. Two-dimensional rigid-body collisions with friction. *Journal of Applied Mechanics*, 59(3):635–642, 1992.

[97] Tanja Weber. *Inverse Differential Kinematics: Exploiting a Dynamically Unconstrained Joint*. Master thesis, Technische Universität München, München, 2017.

[98] J. Zachary Woodruff and Kevin M. Lynch. Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. In *IEEE International Conference on Robotics and Automation*, pages 4066–4073. IEEE, 2017.

[99] Arturo Zavala-Rio and Bernard Brogliato. On the control of a one degree-of-freedom juggling robot. *Dynamics and Control*, 9(1):67–90, 1999.

[100] Hajer Zghal, Rajiv V. Dubey, and J. A. Euler. Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. In *IEEE International Conference on Robotics and Automation*, pages 1006–1011. IEEE Comput. Soc. Press, 1990.

[101] Jun Zhang, Karl Henrik Johansson, John Lygeros, and Shankar Sastry. Zeno hybrid systems. *International Journal of Robust and Nonlinear Control*, 11(5):435–451, 2001.