Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Advances for Agile and Reactive Dynamic Manipulation in Human–Robot Collaboration

Gerold Thomas Huber

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

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:
Prof. Gordon Cheng, Ph.D.

Prüfende der Dissertation:
1. Priv.-Doz. Dr.-Ing. habil. Dirk Wollherr
2. Prof. Dr. Johannes Gerstmayr

Die Dissertation wurde am 15.06.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 18.02.2022 angenommen.

This work is dedicated to my beloved wife Luz Amanda
for her unconditional support and endurance during this journey.

Preface

The unbroken fascination of autonomous robots was unquestionably the motivational driver throughout my entire studies, from undergraduate to post-graduate. At the Chair of Automatic Control Engineering (LSR, dt. Lehrstuhl für Steuerungs- und Regelungstechnik), at the Technical University of Munich (TUM), I got the unique opportunity to be involved in a project exploring *human–robot collaboration* in industrial contexts. I thank Dirk and Martin for believing in my abilities to carry this project together with Volker, Khoi, Oz and Zengjie. Together we formed a truly diverse project team that was supervised with boundless academic and creative freedom. Although project life of course had its ups and downs, I am truly grateful for the grown friendship with each one of the project members on a private level. The same holds true for Sebastian and Salman in my second big project *HR-Recycler*, as well as its spin-off project *SCHROTT*. A special thanks also goes to Christoph Meier from the Institute for Computational Mechanics at TUM, for all the fruitful discussions and reigniting my motivation for the elastic beam modelling. I further want to thank Markus for introducing me to the teaching best practices and the excellent collaboration on our course *Control Systems*, Stefan for the scientific and private discussions on shared commuting train rides and private occasions. My long-term office mate Fangzhou and short-term office mate Alex for the fruitful and enlightening mathematical discussions. Tim, Jonas, Alex and the rest of the kicker fans for the important kicker breaks. Larissa and Miruna for all their patients at my bureaucratic questions – but even more importantly – the enjoyable coffee breaks. But the one LSRian I thank the most is Volker. Without a doubt, he was there for me from the day of my PhD job interview to the very end. Thank you for the uncountable hours of shared passion and successes as well as frustration. It was truly a wild ride and I am grateful for our close friendship.

Fortunately, I was also able to free up some left for my private life outside of LSR, where I want to thank my family for their unconditional support. But foremost I am grateful to have found time to marry my beloved "*Shine*" and even start our own little family with our daughter Néia and angel Liam. I am endlessly happy that you became the center of my life. Thank you for *shining* up every single day of it.

Munich, January 2021 Gerold Huber

Acknowledgments

The research leading to these results has received funding from two distinct sources over the lifetime of this thesis. The first part, i.e. the "*human–robot collaboration in industrial contexts*" project, has partially been supported by the SIEMENS AG. The second project I was involved with, i.e. "*HR-Recycler - Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment*", received funding from the Horizon 2020 research and innovation programme under grant agreement №820742. Its spin-off project "*SCHROTT - Safety for a Collaborative Human-RObоT recycling Team*" has further received funding, as an awardee within the COVR project, from the European Union's Horizon 2020 research and innovation programme under grant agreement №779966.

Zusammenfassung

Das Erreichen oder sogar Übertreffen der menschlichen Vielseitigkeit bei Robotermanipulationsaufgaben, gehört zu den ultimativen Zielen der modernen Roboterforschung. Herkömmliche Industrieroboter übertreffen Menschen seit ihren Anfängen in Bezug auf Präzision und Ausführungsgeschwindigkeit. Es ist jedoch immer noch unverständlich, mit welcher Leichtigkeit Menschen in der Lage sind Objekte dynamisch zu manipulieren und dies auch in den unsichersten dynamischen Umgebungen – z.B. in Teams mit anderen Menschen. Beide Aspekte, die Manipulation unter Ausnutzung dynamischer Effekte als auch sich dynamisch ändernde Umgebungen, bieten immer noch eine Fülle offener Forschungsthemen. Da eine vollständige Abhandlung in dieser Forschungsphase außer Reichweite ist, werden in dieser Arbeit zwei besondere Fortschritte in Richtung dieses ehrgeizigen Ziels herausgearbeitet. Der erste Teil beschreibt ein Schema für agile und reaktive Roboter-Endeffektorbewegungen, wie es in hochdynamischen Umgebungen wie der Zusammenarbeit zwischen Mensch und Roboter erforderlich ist. Er besteht aus einem Online-Trajektoriengenerator, der direkt im kartesischen 3D-Aufgabenbereich eines Endeffektors agiert, sowie der Entwicklung hocheffizienter Ausdrücke in geschlossener Form um diese Endeffektorbewegungen auf Gelenke und schließlich eine Manipulierbarkeitsmetrik abzubilden – für den speziellen Fall serieller Kinematiken mit 7 Freiheitsgraden. Die Kombination der beiden Ansätze mit Suchalgorithmen der Graphentheorie führt zum ersten Mal zu einer global optimalen und online-fähigen Redundanzauflösungsmethode, die in der Lage ist lokalen Suboptima zu entkommen und vorhergesagte Singularitäten zu vermeiden. Der zweite Teil dieser Arbeit trägt zum Bereich der dynamischen Manipulationsfähigkeiten bei, welcher auf eine angemessene Modellierung der natürlichen Dynamiken beruht. Er beinhaltet die Ausarbeitung eines kompakten – jedoch geometrisch exakten – analytischen Balkenmodells für beliebig große planar elastische Verformungsdynamiken, sowie einen entsprechenden Simulationsansatz. Während das in Teil I entwickelte Trajektoriengenerierungsschema einem Roboter die sicherheitskritische Beweglichkeit zur Aufrechterhaltung der Reaktionsfähigkeit ermöglicht, dient das in Teil II ausgearbeitete Balkenmodell als Ausgangspunkt für die zukünftige Erforschung dynamischer Manipulationsfähigkeiten verformbarer Objekte.

Abstract

Reaching or even surpassing human versatility in robotic manipulation tasks is among the ultimate goals for modern robot research. Conventional industrial robots outperform humans in terms of precision and execution speed since their early days. Yet it is still incomprehensible with what ease humans can dynamically manipulate objects and doing so even in the most uncertain dynamic environments – e.g. in teams with other humans. Both aspects, manipulation exploiting dynamic effects as well as dynamically changing environments, have still an abundance of open research topics to offer. Because a full treatise is out of scope at this stage of research, this thesis elaborates on two distinct advances towards this ambitious goal. The first part describes a schema for agile and reactive robot end-effector motions, as required in highly dynamic environments such as human–robot collaboration contexts. It comprises an online trajectory generator that directly operates in the 3D Cartesian task space of an end-effector, and the development of highly efficient closed-form expressions to map these end-effector movements to joints and eventually a manipulability metric – for the particular case of serial 7-DOF kinematics. Combining the two approaches with graph search techniques, results for the first time in a globally optimal online redundancy resolution approach, that can escape local sub-optima and avoid predicted singularities. The second part of this thesis contributes to the field of dynamic manipulation skills, which relies on proper modelling of the natural dynamics. It contains the derivation of a concise – yet geometrically exact – analytical beam model for arbitrarily large planar elastic deformation dynamics and a corresponding simulation approach. While the trajectory generator scheme developed in Part I enables a robot to maintain the safety-critical agility for providing reaction capacity at all times, the elaborated beam model in Part II serves as a starting point for future exploration of dynamic manipulation skills of deformable objects.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

BGL	Boost Graph Library	82
CDF	cumulative distribution function	63
CLIK	closed-loop inverse kinematics	43
CPU	central processing unit	42
DAG	directed acyclic graph	77
DOF	degrees of freedom	114
FEM	finite element method	115
FIR	finite impulse response	18
FK	forward kinematics	47
GPU	graphics processing unit	114
HRC	human–robot collaboration	113
IK	inverse kinematics	114
MPC	model predictive control	83
M	manipulability	47
ODE	ordinary differential equation	97
OTG	online trajectory generator	114
PDE	partial differential equation	115
PDF	probability density function	63
PR	parameter requirements	50
PTP	point-to-point	114
RRG	rapidly-exploring random graph	18
SAC	sequential action control	18
SIMD	single instruction multiple data	42
SQP	sequential quadratic programming	18
S-R-S	spherical-revolute-spherical	46
TCP	tool center point	114
TSP	task space projection	47
TSS	task space surjection	47

"There are an endless number of things to discover about robotics. A lot of it is just too fantastic for people to believe."
— Daniel H. Wilson (2011)

1 Introduction

Manipulation is the oldest application that robots were commercially used for. With the famous UNIMATE, designed by George Devol in 1954, being the first programmable robot [DJ54]. UNIMATE was industrially used by General Motors in their assembly lines as early as 1961. A commercial advertisement proclaimed

> *UNIMATE: a machine that can reach out to seven feet and perform a multitude of tasks in factory or laboratory as skillfully as a man but without getting tired.* *[Mov67]*

with video footage of the robot relocating different objects and eventually pouring a person at a bar a glass of whisky from a bottle. While arguably being a tough claim to make at that time, even modern robots are far from reaching human skillfulness in terms of manipulation capabilities. Even more so, if they operate in close vicinity of another human, such as the suggested robot bartender scenario. Developing a versatile manipulation skill set is probably an everlasting branch of robot research. Yet besides the actual task that is ought to be achieved, the presence of humans forms a special environmental setting that adds further requirements to the algorithms in terms of adaptability and safety.

Manipulation Conventional robot tasks are still often limited to *prehensile* object manipulation i.e. manipulation tasks where the object – or potentially a tool – is restricted in all its degrees of freedom (DOF). Assuming that the object dynamics itself can be neglected w.r.t. the manipulator dynamics, the object follows the motion of the manipulator without the need for additional considerations. This holds true for e.g. rigid objects whose mass is much smaller than the manipulator. It is the fundamental assumption of many conventional *pick and place* tasks. However, the field of robot manipulation is much richer if the assumptions are relaxed accordingly.

The classical taxonomy of *manipulation* differentiates according to the considered phenomena. The most basic category *kinematic manipulation* solely depends on the geometry of the machinery. Practically, however, not considering any force interrelation is limited to relatively slow motions and negligible object mass. Kinematic models are often used in e.g. path planning and collision avoidance algorithms. The remaining manipulation categories consider different sets of forces as causes for motion, thus falling into the field of *kinetics* – in the sense of classical mechanics – reaching from static to dynamic manipulation. In *static manipulation* only static forces, e.g. contact and applied load forces, are considered. Note that this form of manipulation is restricted to the above mentioned prehensile object manipulation, as it does not allow any relative movement between the tool and object. Hence, static models can be used for stability analysis of static grasp poses or object assemblies. Adding quasi-static forces such as sliding friction and impact forces to the equations, leads to *quasi-static manipulation* and builds the first category that does allow for non-prehensile manipulation, e.g. pushing and sliding, while still assuming static equilibrium at all times. A less prominent sub-division towards dynamic manipulation is the category of *quasi-dynamic manipulation.* It does consider first order inertia terms i.e. linear accelerations, but neglects velocity terms i.e. Coriolis forces, centrifugal forces and Euler forces. This category thus contains manipulation skills where none or only slow rotary movement occurs. Unconstrained *dynamic manipulation* with high manipulation speeds eventually requires consideration of the full equations of motion. See [Mas01] for corresponding in-depth analyses of each manipulation category.

Fully dynamic manipulation is arguable the most challenging class, yet offers the largest versatility. It does not depend on a perpetual static force equilibrium, but allows for periods within the execution of a manipulation skill that are not fully controlled. Instead, the manipulation depends on the exploitation of so-called *natural dynamics* that act on the machinery as well as the object. These can be of *extrinsic* nature such as environmental contact, friction and gravity, or *intrinsic* nature like inertia and elasticity.

In contrast to static manipulation skills, that could be halted at any instance due to perpetual static force equilibrium, dynamic manipulation skills depend on the interrelation of forces as well as motion and thus can only be treated in their entirety. Doing so, results in agile manipulation capabilities that can e.g. extend the workspace of the robot via throwing and catching [SGB15, AN20], as well as enrich the reachable states by abandoning static equilibria constraints [DXCR93, Pha14] or considering large deflections of a deformable object [HW16]. Dynamic manipulation also allows to achieve complex behavior with simple robot end-effectors e.g. pushing and sliding [LM96, HR20] or even in-hand manipulation with simple grippers [DRP$^+$14, SWUL17]. Albeit interesting case studies concerning dynamic manipulation have been conducted since the early 90's [ML93, Kod93], progress – especially in the field of non-prehensile dynamic manipulation – has been relatively slow. Besides earlier reviews given by Mason and Lynch [Mas99, LM99], two authors that fundamentally characterized this field of research, a more recent survey by Ruggiero et al. is given in [RLS18].

While dynamics due to extrinsic forces – contact, friction, and gravity – as well as intrinsic inertia forces received attention in the research body of robot manipulation, exploiting elastic dynamics in that context is still an underdeveloped field of research.

Environment The manipulation task itself, however, is not the only component that defines necessary requirements on the robot behavior. One of the main classification factors used for describing environments, differs between a *static environments* in which only actions of the robot modify the setting, and a *dynamic environment* where the changes without robot intervention may occur. In the simplest case, this can be objects moving on a conveyer belt. But even this simple case requires sophisticated strategies if uncertain behavior in such environments requires adaptive behavior in real-time, cf. [SWW$^+$19]. Although industrial settings could have been conventionally considered controlled static environments, the vision of introducing collaborative teams consisting of robots as well as human co-workers, represents

a highly dynamic environment from the robot point of view, in which humans are arguably the most unpredictable component. The exploration of these promising human–robot collaboration (HRC) scenarios is in fact a very active field of robot research. An estimation based on data from google scholar, shows an exponential growth of yearly published research papers containing the term "human–robot collaboration" within the last decade – roughly doubling every two years and counting 2730 papers published in the year 2020. Reviews of this immense body of research are given in [BWB08, AZI$^+$18, VPLS18], as well as [MMZ$^+$19]. The latter focussing on industrial manufacturing application.
A central premise for operating in HRC teams, is the creation of a safe environment for the human as well as the robot. As stated in the arising international safety standard [ISO16], this safety aspect relies on force limitation for static, along with the limitation of possible energy transfer in case of dynamic collision. Although it is crucial to employ safe robot control strategies in case of contact, cf. Haddadin [Had15], collisions certainly ought to be avoided whenever possible. Vicentini [Vic21] outlined how to implement the upcoming safety standards in a HRC scenario.
While the above mentioned publications focus on certain aspects of robotic manipulation in HRC environments, Billard and Kragic [BK19] give an overview of current trends and challenges in the broader field of robot manipulation reaching from environmental perception to robot control.

Although having great potential, dynamic manipulation skills are not yet considered in such dynamic environments. A combination of dynamic manipulation and its application in a dynamic HRC environment, forms the realm towards which this work summarizes the author's scientific contributions.

1.1 Positioning of the Thesis Content in the Overall Context

Of course, by no means does this thesis describe a complete treatise of this highly complex topic. Its content represents contributions to two distinct gaps in the literature. Figure 1.1 illustrates how this work relates to the outlined overall context of dynamic manipulation in dynamic environments. The first part deals with robot motion generation tailored to HRC scenarios and promotes the execution of manipulation skills designed in the end-effector task space. The second part intends to apply leverage to exploiting large deformation dynamics in dynamic robot manipulation.

Agile and Reactive Robot End-Effector Motions Seeking inspiration for manipulation skills from humans suggests that the robot movements should be fundamentally planned in the *free end-effector task space.* However, the end-effector trajectories eventually require a mapping to robot joint trajectories and – depending on the robot configuration – these joint trajectories may violate hardware constraints. The *agility* of the robot thus strongly depends on adequate *joint mapping* to preserve its motion capabilities.
But planning in task space addresses two more issues. The *legibility* of the robot motions, i.e. the observer's inference of the correct intent from the ongoing motion [Dra15], is naturally improved when planning in the end-effector space rather than joint space. The more fundamental challenge, though, is ensuring *safety* for the environment – foremost human co-workers but also for the robot itself. It further demands *reactiveness* and adaptive behavior, given

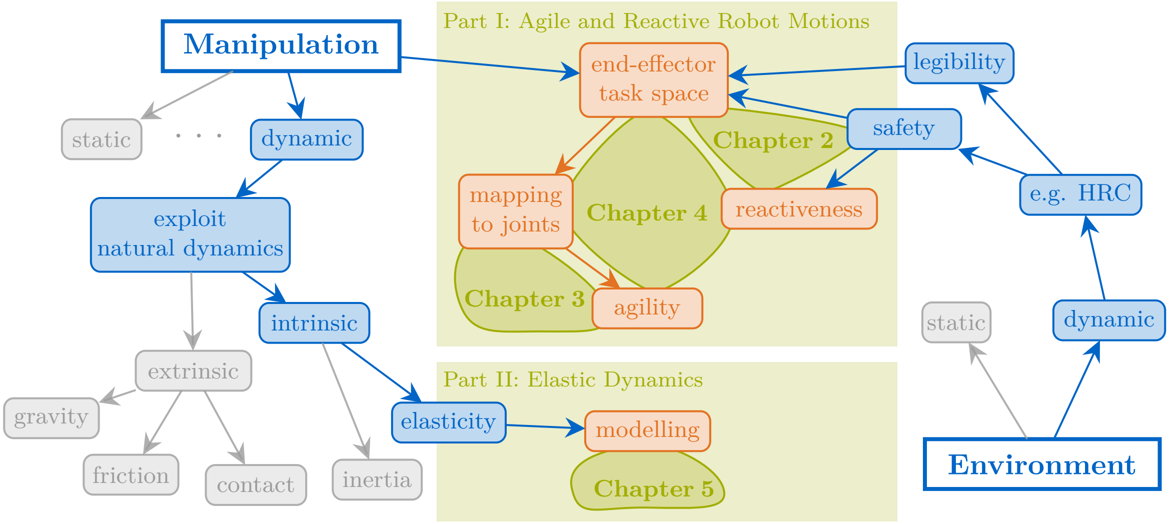


Figure 1.1: Thesis overview and positioning of the content in the overall context.

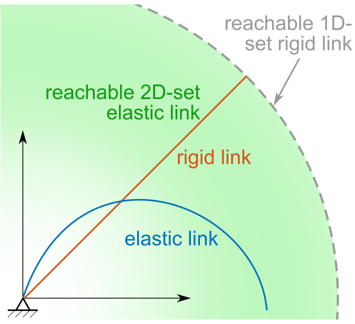


Figure 1.2: Comparison of the reachable 1D path (gray color) of a rigid object and the reachable 2D target set (green color) under consideration of large elastic deformations

the dynamic nature of an HRC environment. Hence, online generation of all trajectories with the capability of reactive motions in unforeseen directions is necessary – rather than possibly pre-calculated trajectory executions. While this thesis does not focus on the development of manipulation primitives per se, it provides fundamental algorithms that generate trajectories directly in the end-effector task space and comply with safety regulations as formulated in the emerging HRC standard [ISO16].

Elastic Dynamics Exploiting natural dynamics is key for agile and versatile robot manipulation skills. However, we propose that the intrinsic potential of large deformation elasticity has yet unexplored but interesting manipulation skills to offer. Especially for the case of dynamic large deflections. This is motivated by the example of manipulating a planar elastic beam, as illustrated in Figure 1.2. One end of the object is fixed in position and angle by e.g. a robot end-effector, while the other end is free. In contrast to its rigid counterpart, whose free end can only reach points on the 1D contour of the circle, consideration of the full elastic potential of the deformable link extends the set of reachable points to the full 2D area. Other ideas for exploiting intrinsic elasticity are the dynamic handling of wires and tubes, or reaching higher kinetic end-effector energies by controlled release of previously built up elastic potential. Development of robot manipulation skills that truly exploit elastic dynamics, however, requires appropriate dynamic models that allow for efficient evaluation of control strategies along with stability analysis.

1.2 Fundamental Themes and Challenges

The distinct chapters in this thesis all have their unique set of challenges. However, there are some common themes among all chapters that can be characterized with the following notions:

End-Effector Movement in Task Space Exploiting natural dynamics for dynamic manipulations skills requires generating end-effector trajectories directly in the kinematic-agnostic Cartesian task space. Yet also static manipulation benefits from generation in task space. It is a more natural approach than planning directly in joint space, and is thus more intuitive and legible for a human to understand and anticipate. However, the rigorous treatment of planning 3D rotations is mathematically more complex, due to their nonlinear and highly interconnected nature.

Agility Eventually, the end-effector movement has to be generated by a robot kinematics. While pre-planned trajectories may be properly positioned in the robot task space in order to not violate hardware constraints, providing reactive behavior is only possible if the robot stays agile at all times. Redundancy of a robot kinematics can be exploited to further preserve maneuverability in terms of a metric, however, these metrics result in complex expressions and are thus difficult to optimize.

Reactive Behavior Reactive behavior requires the online capability of algorithms. This disqualifies a large body of optimization-based strategies that rely on iterative convergence of the solution. Maximal reactiveness can only be achieved, if all results are available within a single control cycle of the robot hardware i.e. typically 1 ms.

Analytical Formulations To achieve the best quality for solutions and allow for rigorous analysis, it is further preferential to not rely on numeric approximations but exact solutions. This is especially challenging for the nonlinearly coupled expressions of 3D rotations, as well as the description of large continuous elastic deformations.

1.3 Scientific Contributions

The contributions of this thesis to the scientific body were all previously published in international journals and conference proceedings. The following paragraphs summarize the main contributions of each chapter.

Online Trajectory Generation (Chapter 2) The first main contribution treats the problem of online trajectory generator (OTG) in the special context of HRC and was previously published in [HGW17] and [HW20]. The chapter introduces the Magnus expansion to OTG problems, for rigorous treatment of time evolution in 3D orientations. It further proposes the first real-time capable OTG that constraints translational and rotational magnitude values of 3D pose dynamics in a singularity-free formulation. This is fundamental to comply with the emerging HRC safety standard ISO/TS 15066 [ISO16].

Manipulability (Chapter 3) As a second main contribution, we propose an efficient closed-form manipulability treatment directly from task space description, for the popular serial 7-DOF robot kinematics [HW19]. The very high computational efficiency, achieved through closed-form expressions that allow for vectorized evaluation, enables extensive analysis of the task space manipulability mapping, as well as its application in global optimization scenarios.

Redundancy Resolution (Chapter 4) Third, we present the first online-capable globally optimal redundancy resolution of a robot kinematics along an end-effector task space trajectory [HW21]. It is capable of traversing between local optima in the null-space and avoid singular configurations. This is achieved by exploiting our methods on the OTG in task space from Chapter 2 as well as the effective manipulability evaluation from Chapter 3, and combining it with graph search techniques.

Beam Modelling (Chapter 5) As a fourth main contribution, we lay the foundation for future dynamic manipulation skill development by proposing a concise and geometrically exact planar beam model for arbitrarily large elastic deformation dynamics in [HWB21]. The model is stated in its weak form as a single partial differential equation (PDE), as well as its strong form for finite element method (FEM) simulations.

1.4 Thesis Outline

The remainder of the thesis is organized by grouping the chapters into two distinct parts, as discussed in Section 1.1 and illustrated in Figure 1.1.

Part I contains the chapters regarding agile and reactive robot end-effector motions. Therein, Chapter 2 discusses an OTG algorithm directly in end-effector task space. Use-cases tailored to HRC-specific scenarios are highlighted to demonstrate maximally reactive trajectory generation while ensuring human safety. While the treatment in this chapter is mostly considering motions in the end-effector task space thus being independent of a specific robot kinematics, Chapter 3 focuses on preserving maximal maneuverability of the manipulator for the specific case of a 7-DOF serial robot kinematics. It further outlines applications in global optimality treatments, considering fundamental robot application questions e.g. optimal mounting poses of the manipulator, as well as optimal exploitation of the kinematic redundancy. Chapter 4 demonstrates how the developed methods of the previous two chapters can be combined, to achieve an online-capable global redundancy resolution framework based on graph search techniques.

Part II of this thesis, deals with the mathematical modelling of intrinsic elastic dynamics, with a focus on providing analytic models for dynamic manipulation. It consists of Chapter 5, where a geometrically exact model for arbitrarily large deflections of an elastic beam is proposed.

Both thesis parts contain their respective discussion of nomenclature and list of symbols in their preamble. Further, all chapters contain their corresponding state of the art along with chapter-specific contributions and conclusions. Chapter 6 gives an overall conclusion about the potential of the work developed in this thesis, as well as an outlook for future research opportunities.

Part I

Agile and Reactive Robot End-Effector Motions

Nomenclature of Part I

Scalars are written in plain lower case, vectors in bold-face lower case. Matrices are bold-face upper case, while plain upper case symbols refer to coordinate frames, mathematical spaces and sets. Subscript annotations are reserved for index notation of multi-dimensional variables, whereas superscript annotations are part of the variable specification.

Coordinate transformation matrices are written as $\boldsymbol{R}_{kj}$ with 2 indices and are read *from right to left*, e.g. $\boldsymbol{R}_{43}$ transforms the coordinate system from body-fixed frame of joint q_3 to joint q_4. Whereas vector indices are read *from left to right*, and their reference frame is written as left-hand side subscript. The notation ${}_{B}\boldsymbol{r}_{SW}$, thus describes a vector $\boldsymbol{r}$ pointing from shoulder S to wrist W, expressed in base frame B. If a vector does not have a lower left index, it always refers to the base B.

Also note that we omit explicit listing of function parameters whenever it is clear from the context, to not unnecessarily clutter the notation. A list of the most frequently used variables in this work is given.

Scalars and Vectors

$\boldsymbol{a}$	translational acceleration vector
$\boldsymbol{\alpha}$	rotational acceleration vector
$\boldsymbol{\mathit{\Lambda}}$	total acceleration vector consisting of $\boldsymbol{a}, \boldsymbol{\alpha}$
$\boldsymbol{e}$	error vector
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J}$	manipulator Jacobian
k	discrete time variable
λ	null-space arm angle of 7-DOF kinematics
μ	manipulability metric
$\boldsymbol{\omega}$	rotational velocity vector
$\boldsymbol{p}$	parameter vector
$\boldsymbol{q}$	robot joint angle vector
$\mathfrak{q}$	3D orientation as a unit quaternion
$\boldsymbol{r}$	translation position vector
$\boldsymbol{R}$	3D rotation matrix
σ	saturation factor
t	continues time variable
$\mathit{\Delta} t$	sampling time
T	time span

$\boldsymbol{v}$	translational velocity vector
$\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$	Cartesian base vectors of a coordinate system
$\boldsymbol{x}$	state vector
$\boldsymbol{z}$	3D pose

Indices

$\bar{(\cdot)}$	temporary variable
${}_C\boldsymbol{r}_{AB}$	vector from point A to B, expressed in the reference coordinate system C
$\boldsymbol{R}_{DC}$	Rotation from reference coordinate system C to D
des	desired
goal	goal
max	maxima
min	minima
pred	prediction
rot	rotation
sat	saturation
tot	total
tra	translation

Descriptors of 7-DOF Serial Robot Kinematics (cf. Figure 3.3)

1 - 7	joint locations 1 to 7
I	world-fixed inertial frame
B	base/origin of the robot kinematics
S	shoulder of robot kinematics, i.e. joint 2
E	elbow of robot kinematics, i.e. joint 4
W	wrist of robot kinematics, i.e. joint 6
T	tool of robot kinematics, i.e. end-effector

Spaces and Sets

$\mathcal{A}$	set of admissible parameters
$\mathbb{C}$	space of complex numbers
$\mathcal{C}$	set of OTG constraints
$\mathcal{G}$	graph of manipulability measures of null-space along a trajectory
$\mathbb{H}$	algebra of Hamilton's quaternions
$\mathcal{I}$	set of axes intersecting points on the kinematics $\{S, E, W, T\}$

$\mathcal{M}$	discrete grid of manipulability measures in null-space along a trajectory
$\mathbb{N}_+$	space of positive natural numbers i.e. without 0
$\mathcal{P}$	space of new task and null-space parameterization
$\mathcal{Q}$	space of admissible joint configurations
$\mathbb{R}^n$	n-dimensional space of real numbers
$\mathbb{R}_+$	space of positive real numbers
$\mathcal{S}$	set of saturation factors
SO(3)	special orthogonal group (3D rotations)
SE(3)	special Euclidean group (3D rigid motions)

Operators

$(\cdot) \times (\cdot)$	vector cross product
$(f \circ g)(a, b)$	function composition $f(g(a, b))$
$[A, B]$	commutator $[A, B] := AB - BA$
$\det(\cdot)$	determinant of a matrix
$\lceil \cdot \rceil$	ceiling operator that maps to the least integer greater than or equal to
$\lVert \cdot \rVert_2$	2-norm of a vector
$[\,\cdot\,]_\times$	skew operator $[\,\cdot\,]_\times : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$
$[\,\cdot\,]_\vee$	inverse skew operator $[\,\cdot\,]_\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$

"Even a boxer on awakening from a knockout punch knows that the impulse vector applied to his chin has a physical significance independent of any coordinate system used to describe it."

— Stuart S. Antman, in *'Nonlinear Problems in Elasticity'* (2005)

2 Online Trajectory Generation for Safe HRC

This chapter was previously published in [HW19].

With the increasing demand for humans and robots to collaborate in a joint workspace, it is essential that robots react and adapt instantaneously to unforeseen events to ensure safety. Constraining robot dynamics directly on SE(3), i.e. the group of 3D translation and rotation, is essential to comply with the emerging human–robot collaboration safety standard ISO/TS 15066. We argue that limiting coordinate-independent magnitudes of physical dynamic quantities at the same time allows more intuitive constraint definitions. We present the first real-time capable OTG that constrains translational and rotational magnitude values of 3D translation and 3D rotation dynamics in a singularity-free formulation. Simulations as well as experiments on a hardware platform show the utility in human–robot collaboration contexts.

2.1 Introduction

Enabling humans and robots to physically work together on cooperative tasks in close distance is a very active area of current robotic research. In such HRC scenarios, a trajectory generator must not only assure human safety but at the same time respect human comfort, in order to increase acceptance of the robot by human co-workers. The former – safety – asks for a real-time capable trajectory generator that reacts instantaneously to sensor inputs or a change in constraints. This requires evaluation of new set-points in every embedded low-level iteration cycle at high frequencies. While literature offers various solutions for OTGs with axis-specific constraints on different derivative orders, the latter – comfort – favors constraining velocities and accelerations directly in the most intuitive context for the human. Namely 3D translation as well as 3D rotation in their geometric sense, i.e. vectors with an orthonormal bases. Rather than limiting dynamics of the coordinate-wise components independently, human presence further suggests restricting end-effector movement dynamics in their absolute values.

Restricting the magnitude of translational velocity is in particular essential for incorporating safety standards in HRC according to ISO/TS 15066 [ISO16]. Although this international standard is still under development, it serves as a guideline on how safety during collaborative operations must be provided. Possible methods listed in this standard are *safety-rated monitored stop*, *hand guiding*, *speed and separation monitoring* as well as *power and force limited collaborative operation*. Especially for collaboration in close distance only the latter is an appropriate option. Within this method of operation, contact events are categorized in quasi-static and transient contact. The former includes clamping or crushing situations and mainly addresses the robot controller. The latter treats dynamic impact hazards, which can be actively considered in OTGs. Safety is provided by limiting the transferred energy during impact, according to different human body regions. Given the effective mass of the two-body system, one can derive the maximal allowed relative speed between the robot and the human body region. Detailed calculation guidelines are outlined in [ISO16].

Considering these facts we propose the following requirements for an OTG in HRC contexts:

R1: Translation and rotation dynamic quantities are constrained in their magnitudes.

R2: Rotation dynamics refer to geometric angular quantities.

R3: All degrees of freedom are synchronized in their motions.

R4: Constraint-conform trajectories are directly forwarded without delay.

R5: Iteration cycle times below 1 ms must be guaranteed.

Fulfilling these requirements is not addressed in current state-of-the-art OTGs.

2.1.1 Related Work

Current OTGs [CFK16] are able to make robotic systems robust against infeasible trajectory inputs due to e.g. step inputs, improper path transitions caused by a potential higher-level path planner, or possible communication delay fluctuations. Online capability also enables instantaneous reaction to sudden unforeseen events [KW10]. However, authors usually either consider trajectory generation in joint space while moving close to hardware limits to achieve time optimal solutions, or extend these approaches to end-effector movements by treating the

Requirement		*R1*	*R2*	*R3*	*R4*	*R5*
Direct approaches [KW10, BSHA08, HWR08, KKAK13, ETS14] [LAS16b, LAS16a, Krö11, RIS13, Llo98]		✗	(✓)^a	✓	✗	✓
Indirect approaches	FIR Filter based [BM12, BMM16, BBG16]	✗	✗	✓	✗	✓
	Sliding Mode based [GB10, BG14, Bia17]	✗	✗	(✓)^b	✓	✓
Optimization based [AORJ15, DWLW17, KMP18] [GLS17, LLM08, WSK16]		✗	✓	✓	✓	✗
Our approach		✓	✓	✓	✓	✓

[a] only true for [Llo98] [b] only true for [Bia17]

Table 2.1: Comparison to the state-of-the-art w.r.t. the defined requirements

translational and rotational DOFs as independent axes in $\mathbb{R}^n$. Due to the demand for instantaneous reaction to unforeseen events (*R5*), pure path planners without temporal information as well as pure path parametrization methods are not considered in this short review. We classify the relevant OTGs approaches into three major groups:

Direct approaches define a trajectory profile and pre-compute the whole trajectory to the target state. Most trajectory generators are based on piecewise polynomials as they can be easily designed for arbitrarily often differentiable trajectories. Many approaches, however, are restricted to zero dynamic start and/or end conditions, e.g. [BSHA08, HWR08], which entails a constant time delay when feeding the algorithm with already feasible trajectories. Kröger et al. [KW10] provided a basic concept for OTGs and also introduced a classification scheme. Katzschmann et al. [KKAK13] extended the approach to regard the entire robot dynamics. Ezair et al. [ETS14] proposed an iterative approach, that uses recursive S-curve polynomials to generate trajectories of arbitrary order with general initial and final conditions. Another iterative algorithm that considers a sampled input trajectory was proposed by Lange et al. [LAS16b] and later improved for fast trajectories in [LAS16a]. Their work focuses on path-accurate OTG. These approaches are designed for application in joint space. Although the joint trajectories could be derived from an analogous Cartesian trajectory, only constraints in joint space are considered. Kröger expanded the ideas from [KW10] to straight motions in 3D Euclidean space in [Krö11] by considering orientation in the minimal representation by Euler angles. However, the derivations of these values do not lead to geometric angular velocities, but are merely of analytic nature as they represent concatenated velocities. This important difference will be further discussed in Section 2.5.2.

Rymansaib et al. [RIS13] propose the usage of exponential functions rather than polynomials for trajectory generation. This allows consideration of hardware constraints as well as generation of trajectories with continuous jerk. They show that the exponential trajectory generation results in higher tracking accuracy than common S-curve velocity methods. Nevertheless, their approach is also designed for pre-planned point-to-point (PTP) motions with zero target velocities/acceleration and therefore suffers from time delays as well, which conflicts with *R4*. Moreover, none of the above mentioned direct methods can fulfill *R2*. The only to the authors known OTG that does formulate geometric angular velocity constraints on SO(3), i.e. the group of 3D rotations, is an unpublished work by Lloyd [Llo98]. He reduces the 3D problem to two 1D sub-problems by transformation to a locally radial and thus easily solvable coordinate system. Due to that decomposition, however, the approach does not comply with *R1* and further does not allow desired velocity set-points, necessary for *R4*.

Indirect approaches consider the trajectory generation as dynamic control system problem, i.e. a chain of integrators. Filter-based techniques were adopted by Biagiotti et al. [BM12, BMM16] and Besset et al. [BBG16]. The trajectory generators proposed by Besset et al. use finite impulse response (FIR) filters to generate jerk-limited profiles out of initially acceleration-limited trajectories. While a pre-generated trajectory is given in advance, it delivers a solution in less than 1 µs and thus is the fastest time-optimal jerk-limited trajectory generator at this time, even for the multi-dimensional case. Biagiotti et al. [BM12] use a chain of FIR filters to smoothen trajectories to have continuous derivatives of arbitrary order. In [BMM16] the strategy is generalized to produce piecewise exponential jerk profiles, to further reduce machine vibrations. Due to their filtering nature, these approaches again result in time delays and thus neither satisfy *R4*. Gerelli et al. [GB10] and Bianco et al. [BG14] proposed a discrete-time filter that incorporates constraints as sliding surfaces. These inspiring works generate time optimal trajectories under consideration of bounded velocity, acceleration and jerk for single-DOF applications. In [Bia17], Bianco proposes a strategy to synchronize multiple DOFs. Besides fulfilling *R4*, the latter extends the approach to further comply with *R3*.

Optimization based approaches rely on numerical solvers. Ardakani et al. [AORJ15] suggest a real-time joint trajectory generator based on model predictive control. Their formulation on $\mathbb{R}^n$ achieves an impressive 200 Hz sampling rate for a 6DOF robot kinematics despite the optimization procedure in every iteration. Dinh et al. [DWLW17] combine sequential action control (SAC) [AM16] with indirect optimization. In numerical optimization frameworks, it is also possible to generate trajectories for hybrid dynamic systems such as the table tennis robot in Koç et al. [KMP18]. While solving the individual optimization problems with sequential quadratic programming (SQP) requires over 1 s computation time, they pre-compute a lookup table from a fixed initial posture that can be used online. However, for a more general trajectory generation problem this approach is infeasible. Gao et al. [GLS17] use a rapidly-exploring random graph (RRG) method, for finding a collision-free trajectory for a quadrotor in complex environments. Note that formulating the kinematics in $\mathbb{R}^n$, which holds e.g. for joint trajectories as well as 3D translation, results in a linear chain of integrators. Optimization of 3D rotation trajectories on the other hand result in highly *R5* and coupled dynamics. Le et al. [LLM08] study the time optimal control problem for SO(3) in a general setting, but only include acceleration constraints. Another algorithm for trajectory generation directly on SE(3) is proposed by Watterson et al. [WSK16]. They use semi-definite programming techniques and also consider obstacles in the environment. Using numerical optimization techniques is usually a limiting factor when hard real-time capabilities as the one in *R5* are required, especially when *R5* constraints and SO(3) dynamics (*R1*)are introduced. None of the above mentioned works was designed for scenarios that impose requirements *R1* and *R2* while being real-time capable as required by *R5*.

Table 2.1 gives a concise summary of the referenced approaches w.r.t. the posed requirements.

2.1.2 Contribution

In this chapter, we develop a task space OTG algorithm that directly applies the special orthogonal rotation matrices for a rigorous treatment of 3D orientations. It is further adapted to unit quaternion representation. Special case singularity treatment as well as consideration of multiple goal states are hence not necessary, in contrast to the often used Euler angle representations.
The main contributions of this chapter are:

1. This is the first OTG algorithm that allows constraining the norm of translational and rotational dynamic quantities, essential to comply with safety standards defined in ISO/TS 15066 [ISO16].

2. We introduce the Magnus Expansion to OTG treatment. This allows calculating solution to the differential equations on SO(3) with high accuracy.

2.1.3 Outline

The remainder of the chapter is organized as follows. The problem of a modular online trajectory generator under requirements *R1-R5* is formally formulated in Section 2.2. In Section 2.3, the approach is first introduced for the translational case and then adapted to the rotational DOFs. Finally, it is shown how the two portions can be synchronized w.r.t. time. For a distinct highlighting of the potential of our algorithm in robotic applications, we outline constraint extensions for HRC scenarios in Section 2.4 and discuss its performance with a comparison w.r.t. to the state-of-the-art OTG algorithms in Section 2.5. We conclude the chapter and outline future directions of development in Section 2.6.

2.2 Problem Formulation

Regarding a modular robot architecture of a complex system, a clear distinction between trajectory generation and robot control is of advantage. Therefore, an interface is needed, that guarantees desired trajectories sent to an arbitrary robot platform stay within defined dynamic constraints. Especially in close distance HRC scenarios it is essential for these constraints to be defined in an intuitive metric, i.e. coordinate-independent magnitudes of translational and rotational speed and acceleration.

Let the end-effector pose of a robot be given in $\mathrm{SE}(3) := \mathbb{R}^3 \times \mathrm{SO}(3)$, that consists of a 3D translational position $\boldsymbol{r} \in \mathbb{R}^3$ and a 3D orientation $\boldsymbol{R} \in \mathrm{SO}(3)$. The special orthogonal group SO(3) is defined as $\mathrm{SO}(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times3} | \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}_{3\times3}, \det(\boldsymbol{R}) = 1\}$, with $\boldsymbol{I}_{3\times3}$ referring to the identity matrix in $\mathbb{R}^{3\times3}$. The system state vector $\boldsymbol{x}$ at time $t_k \in \mathbb{R}$ consists of the full 6D pose, together with translational velocity $\boldsymbol{v} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. It is denoted as the 4-tuple $\boldsymbol{x}(t_k) = \boldsymbol{x}_k := (\boldsymbol{r}_k, \boldsymbol{R}_k, \boldsymbol{v}_k, \boldsymbol{\omega}_k)$. Applying a constant acceleration tuple

$\boldsymbol{\Lambda}_k := (\boldsymbol{a}_k, \boldsymbol{\alpha}_k)$ consisting of translational and rotational acceleration $\boldsymbol{a}_k \in \mathbb{R}^3$ and $\boldsymbol{\alpha}_k \in \mathbb{R}^3$ resp. for a given time span $T \in \mathbb{R}$, the system state advances according to

$$\boldsymbol{x}(t_k + T) := \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{\Lambda}_k, T), \tag{2.1}$$

where $\boldsymbol{f}$ denotes the OTG dependent state propagation function. Assuming a discrete implementation of the OTG running at a sample time $\Delta t \in \mathbb{R}$, we refer to the state at time $t_k + i\Delta t$ with $i \in \mathbb{N}_+$ as $\boldsymbol{x}_{k+i}$.

Given a desired state $\boldsymbol{x}^{\text{des}}$ and the current system state $\boldsymbol{x}_k$, the OTG must solve the problem

$$\underset{\boldsymbol{\Lambda}_k,\ldots,\boldsymbol{\Lambda}_{k+(N-1)}}{\text{minimize}} \quad N \tag{2.2}$$

subject to the state progression

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{\Lambda}_i, \Delta t\right) \tag{2.3a}$$

$$\boldsymbol{x}_{k+N} = \boldsymbol{x}^{\text{des}} \tag{2.3b}$$

and the user-defined dynamic magnitude constraints *R1*

$$\|\boldsymbol{v}_i\|_2 \leq v_{\max} \tag{2.4a}$$

$$\|\boldsymbol{a}_i\|_2 \leq a_{\max} \tag{2.4b}$$

$$\|\boldsymbol{\omega}_i\|_2 \leq \omega_{\max} \tag{2.4c}$$

$$\|\boldsymbol{\alpha}_i\|_2 \leq \alpha_{\max} \tag{2.4d}$$

for all $i \in [k, N-1]$ time steps. The operator $\|\cdot\|_2$ denotes the 2-norm and thus the magnitude of the vectors.

Note that Le et al. [LLM08] solve a similar problem on SE(3) for a reduced set of dynamic constraints. Their numerical examples show computation times of over 90 s for a single trajectory solution, and thus demonstrate the complexity of the problem at hand. Considering the core requirement of every OTG being fast computation, in order to guarantee safety at typical robot sampling frequencies of 1 kHz, the use of iterative numeric optimization algorithms is not appropriate. Accordingly, we relax the requirement of absolute time optimality and accept close-to-optimal solutions under real-time capable iteration cycles $\leq$ 1 ms instead.

2.3 Approach

Considering that the dynamic constraints (2.4) are related to translational and rotational quantities separately, we partition the state dynamics (2.1) into the individual mappings

$$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{\Lambda}_k, T) := \begin{cases} \boldsymbol{r}(t_k + T) & \leftarrow \boldsymbol{f}^r(\boldsymbol{r}_k, \boldsymbol{v}_k, \boldsymbol{a}_k, T) \\ \boldsymbol{v}(t_k + T) & \leftarrow \boldsymbol{f}^v(\boldsymbol{v}_k, \boldsymbol{a}_k, T) \\ \boldsymbol{R}(t_k + T) & \leftarrow \boldsymbol{f}^R(\boldsymbol{R}_k, \boldsymbol{\omega}_k, \boldsymbol{\alpha}_k, T) \\ \boldsymbol{\omega}(t_k + T) & \leftarrow \boldsymbol{f}^\omega(\boldsymbol{\omega}_k, \boldsymbol{\alpha}_k, T) \end{cases} \tag{2.5}$$

according to the state tuple $\boldsymbol{x}_k = (\boldsymbol{r}_k, \boldsymbol{v}_k, \boldsymbol{R}_k, \boldsymbol{\omega})$. The translation dynamics are thus given with $\boldsymbol{f}^r$ and $\boldsymbol{f}^v$, whereas the rotation dynamics consist of $\boldsymbol{f}^R$ and $\boldsymbol{f}^\omega$.

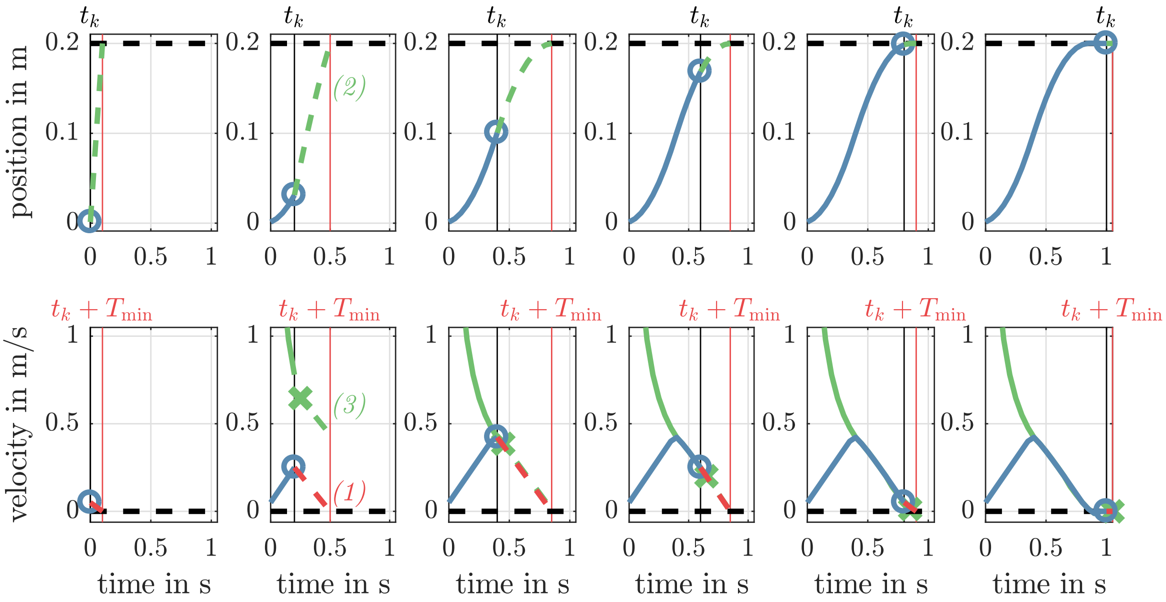


Figure 2.1: Progression of the OTG with initial state $\boldsymbol{r}(0) = 0, \boldsymbol{v}(0) = 0$ and final state $\boldsymbol{r}^{\text{des}} = 0.2, \boldsymbol{v}^{\text{des}} = 0$ at $\Delta t = 0.05$. The progress is given for 6 time instances $t_k = \{0, 0.2, 0.3, 0.5, 0.6, 0.8\}$. Algorithm: *(1)* find $T_{\min}$, the time needed to reach final velocity $\boldsymbol{v}^{\text{des}}$ (dashed red line in velocity plot) *(2)* fit position profile that reaches final position $\boldsymbol{r}^{\text{des}}$ in $T_{\min}$ (dashed green line in position plot) *(3)* find $\boldsymbol{v}_{k+1}^{\text{goal}}$ of corresponding position profile (green mark in velocity plot) *(4)* after incorporating constraints, apply corresponding acceleration for Δt (not shown)

Starting from the basic translational problem in Section 2.3.1, the approach is adapted for the rotational group SO(3) in Section 2.3.2. Based on these formulations, the synchronization over all 6 DOFs w.r.t. time is outlined in 2.3.3. The set of conditions listed in (2.4) from the problem formulation, is extended to additional HRC motivated use-cases in Section 2.4

2.3.1 Translation

For 3D translation, the 3 components can be treated independently if they are defined w.r.t. an orthonormal basis. The desired behavior for the algorithm is a continuously differentiable position trajectory. This is achieved by calculating the next state via integration of the limited acceleration. The discrete system dynamics (2.3a) for the translational case are 3 decoupled double integrators, i.e. for a given position $\boldsymbol{r}$ and velocity $\boldsymbol{v}$ at time t_k, applying a constant acceleration vector $\boldsymbol{a}_k$ for the time T advances according to the mappings

$$\boldsymbol{v}(t_k + T) = \boldsymbol{f}^v(\boldsymbol{v}_k, \boldsymbol{a}_k, T) := \boldsymbol{v}_k + \boldsymbol{a}_k T \tag{2.6a}$$

$$\boldsymbol{r}(t_k + T) = \boldsymbol{f}^r(\boldsymbol{r}_k, \boldsymbol{v}_k, \boldsymbol{a}_k, T) := \boldsymbol{r}_k + \boldsymbol{v}_k T + \frac{1}{2}\boldsymbol{a}_k T^2. \tag{2.6b}$$

The aim of the algorithm is to compute the acceleration such that the position error $\boldsymbol{e}_k^p := \boldsymbol{r}_k^{\text{des}} - \boldsymbol{r}_k$ as well as the velocity error $\boldsymbol{e}_k^v := \boldsymbol{v}_k^{\text{des}} - \boldsymbol{v}_k$ both converge to zero. The minimum time needed to reach $\boldsymbol{v}_k^{\text{des}}$ is achieved, if the acceleration vector $\boldsymbol{a}_k$ in (2.6a) exploits the acceleration constraint (2.4b) and points in the same direction as the velocity error $\boldsymbol{e}_k^v$. Respectively,

$$\boldsymbol{v}_k^{\text{des}} = \boldsymbol{f}^v \left(\boldsymbol{v}_k, a_{\max} \frac{\boldsymbol{e}_k^v}{\|\boldsymbol{e}_k^v\|_2}, \bar{T}_{\min}^{\text{tra}} \right)$$

is solved for

$$\bar{T}_{\min}^{\text{tra}} := \frac{\|\boldsymbol{e}_k^v\|_2}{a_{\max}}. \tag{2.7}$$

As the discrete algorithm will run on a fixed sample time $\varDelta t$, and acceleration is constant in-between iterations, the minimal time needed is in fact the next multiple of $\varDelta t$, denoted by

$$T_{\min}^{\text{tra}} = \max \left\{ \left\lceil \frac{\bar{T}_{\min}^{\text{tra}}}{\varDelta t} \right\rceil, 1 \right\} \varDelta t \tag{2.8}$$

with operator $\lceil \cdot \rceil$ being the ceiling function that rounds up to the next full integer. The maximization operator in (2.8) ensures $T_{\min}^{\text{tra}} \geq \varDelta t$. Requiring to reach the desired velocity $\boldsymbol{v}_k^{\text{des}} = \boldsymbol{f}^v(\boldsymbol{v}_k, \bar{\boldsymbol{a}}_k, T_{\min}^{\text{tra}})$ in the time span $T_{\min}^{\text{tra}}$ is directly used to define the acceleration vector

$$\bar{\boldsymbol{a}}_k = \frac{\boldsymbol{e}_k^v}{T_{\min}^{\text{tra}}}, \tag{2.9}$$

that would result in a synchronous convergence of $\boldsymbol{e}^v(t_k + T_{\min}^{\text{tra}}) = \boldsymbol{0}$ while respecting the acceleration constraint (2.4b).

Applying $\bar{\boldsymbol{a}}_k$ to the position integration mapping $\boldsymbol{f}^r$ and imposing to reach the desired velocity $\boldsymbol{r}_k^{\text{des}} = \boldsymbol{f}^r(\boldsymbol{r}_k, \boldsymbol{v}_k^{\text{goal}}, \bar{\boldsymbol{a}}_k, T_{\min}^{\text{tra}})$ in $T_{\min}^{\text{tra}}$, defines the goal velocity

$$\boldsymbol{v}_k^{\text{goal}} := \frac{\boldsymbol{e}_k^p}{T_{\min}^{\text{tra}}} - \bar{\boldsymbol{a}}_k \frac{T_{\min}^{\text{tra}}}{2}. \tag{2.10}$$

If this velocity is matched, the position error $\boldsymbol{e}^p$ converges to zero together with the velocity error $\boldsymbol{e}^v$ for $T \to T_{\min}^{\text{tra}}$. Advancing $\boldsymbol{v}_k^{\text{goal}}$ for a single iteration leads to

$$\begin{aligned} \boldsymbol{v}_{k+1}^{\text{goal}} &:= \boldsymbol{f}^v(\boldsymbol{v}_k^{\text{goal}}, \bar{\boldsymbol{a}}_k, \varDelta t) \\ &= \frac{\boldsymbol{e}_k^p}{T_{\min}^{\text{tra}}} + \bar{\boldsymbol{a}}_k \left(\varDelta t - \frac{T_{\min}^{\text{tra}}}{2} \right) \end{aligned} \tag{2.11}$$

and is used as the goal velocity at time t_{k+1}.

The second phase of the algorithm handles given constraints by defining two sets of saturation factors $\mathcal{S}^v \subset \mathbb{R}^+$ and $\mathcal{S}^a \subset \mathbb{R}^+$ for velocity and acceleration respectively.

Constraints on the trajectory generator output are incorporated by multiplication with the most restrictive element of the corresponding saturation factor set. The translational speed constraint (2.4a),e.g. is met by the set definition

$$\mathcal{S}^v := \{1, \sigma^v\} \quad \text{with} \quad \sigma^v := \frac{v_{\max}}{\left\| \boldsymbol{v}_{k+1}^{\text{goal}} \right\|_2}. \tag{2.12}$$

Note that all factors in the set are normalized and saturation requires the factor 1 to be part of the set.[1] For the goal velocity $\boldsymbol{v}_{k+1}^{\text{goal}}$ this factorization is applied with

$$\boldsymbol{v}_{k+1}^{\text{sat}} := \boldsymbol{v}_{k+1}^{\text{goal}} \min\{\mathcal{S}^v\}, \tag{2.13}$$

[1] a discussion on how to derive saturation factors is given in Section 2.4

where superscript $(\cdot)$sat denotes a saturated value.

The necessary acceleration vector to reach velocity constraint conform $\boldsymbol{v}_{k+1}^{\text{sat}}$ is found by solving $\boldsymbol{v}_{k+1}^{\text{sat}} = \boldsymbol{f}^{v}(\boldsymbol{v}_k, \boldsymbol{a}_k^{\text{goal}}, \Delta t)$ for the goal acceleration

$$\boldsymbol{a}_k^{\text{goal}} := \frac{\boldsymbol{v}_{k+1}^{\text{sat}} - \boldsymbol{v}_k}{\Delta t}. \tag{2.14}$$

Eventually, a constraint-conform acceleration vector satisfying $\|\boldsymbol{a}_k\|_2 \leq a_{\max}$ is found with

$$\boldsymbol{a}_k^{\text{sat}} := \boldsymbol{a}_k^{\text{goal}} \min\{\mathcal{S}^a\} \tag{2.15}$$

using the set definition

$$\mathcal{S}^a := \{1, \sigma^a\} \quad \text{with} \quad \sigma^a := \frac{a_{\max}}{\left\|\boldsymbol{a}_k^{\text{goal}}\right\|_2} \tag{2.16}$$

analogously to the velocity saturation (2.12).

The output of the trajectory generator at hand is ultimately found by advancing the current state $\boldsymbol{x}_k$ for a sample period

$$\begin{aligned} \boldsymbol{v}_{k+1} &= \boldsymbol{f}^{v}(\boldsymbol{v}_k, \boldsymbol{a}_k^{\text{sat}}, \Delta t) \\ \boldsymbol{r}_{k+1} &= \boldsymbol{f}^{r}(\boldsymbol{r}_k, \boldsymbol{v}_k, \boldsymbol{a}_k^{\text{sat}}, \Delta t). \end{aligned} \tag{2.17}$$

Note that the saturation steps for constraint handling do not conflict with the desired continuous differentiability of the pose trajectories, as velocity and position values are obtained by integration of (2.6) under constant acceleration $\boldsymbol{a}_k^{\text{sat}}$ until the next iteration cycle. Figure 2.1 illustrates the mechanism of the approach for a PTP motion. A concise implementation including all necessary steps is outlined in Algorithm 2.1.

In Section 2.4 additional saturation factors relevant in HRC scenarios are developed. They are integrated by adding elements to the sets defined in (2.12) and (2.16).

2.3.2 Rotation

In the rotational treatment of the OTG lies the main contribution of this chapter. Many different representation forms are known to represent 3D rotations on the special orthogonal group SO(3). In a preliminary work [HGW17] we used Euler angles as they form a minimal description unlike axis-angle, quaternions or rotation matrix representation. This allowed a straight forward analogy between the translational and the rotational treatment. However, the well-known gimbal-lock singularities of this representation form require special treatment. Furthermore, there always exist two sets of angles to describe the same orientation. Exploiting this dual representation allows finding shorter paths in case of point-to-point motions.

Because angular velocity $\boldsymbol{\omega}$ and acceleration $\boldsymbol{\alpha}$ are both defined in $\mathbb{R}^3$ and thus are geometrically decoupled, the algorithm of the translational case can to the most extend directly be adapted for the rotational case. With the exception of $\boldsymbol{f}^r$ defined in (2.10) that incorporates the temporal evolution of the position vector. While position $\boldsymbol{r} \in \mathbb{R}^3$ allows elementary integration of the individual components, the same does not hold for the orientation.

Algorithm 2.1: OTG on $\mathbb{R}^3$ with magnitude constraints

```
Function OTGtranslation
    Data    : current state $(\boldsymbol{r}_k, \boldsymbol{v}_k)$, desired state $(\boldsymbol{r}_k^{\text{des}}, \boldsymbol{v}_k^{\text{des}})$,
              constraints $(v_{\max}, a_{\max})$, sampling time $\Delta t$
    Result  : next state $(\boldsymbol{r}_{k+1}, \boldsymbol{v}_{k+1})$
    /* ===== find new goal velocity ===== */
    $T_{\min}^{\text{tra}} \leftarrow \max\left\{\left\lceil \frac{\|\boldsymbol{v}_k^{\text{des}} - \boldsymbol{v}_k\|_2}{a_{\max}} \right\rceil, 1\right\} \Delta t$ ;    // discretized minimum Time (2.7)
    $\boldsymbol{v}_{k+1}^{\text{goal}} \leftarrow \frac{\boldsymbol{r}_k^{\text{des}} - \boldsymbol{r}_k}{T_{\min}^{\text{tra}}} + \frac{\boldsymbol{v}_k^{\text{des}} - \boldsymbol{v}_k}{T_{\min}^{\text{tra}}} \left(\Delta t - \frac{T_{\min}^{\text{tra}}}{2}\right)$ ;    //  goal velocity (2.11)
    $\mathcal{S}^v \leftarrow$ collectVelocitySaturationFactors($\boldsymbol{v}_{k+1}^{\text{goal}}$,$v_{\max}$) ;
    $\boldsymbol{v}_{k+1}^{\text{sat}} \leftarrow \boldsymbol{v}_{k+1}^{\text{goal}} \min\{\mathcal{S}^v\}$ ;    // saturated goal velocity (2.13)
    /* ===== find goal acceleration ===== */
    $\boldsymbol{a}_k^{\text{goal}} \leftarrow \frac{\boldsymbol{v}_{k+1}^{\text{sat}} - \boldsymbol{v}_k}{\Delta t}$ ;    // goal acceleration (2.14)
    $\mathcal{S}^a \leftarrow$ collectAccelerationSaturationFactors($\boldsymbol{a}_{k+1}^{\text{goal}}$,$a_{\max}$) ;
    $\boldsymbol{a}_k^{\text{sat}} \leftarrow \boldsymbol{a}_{k+1}^{\text{goal}} \min\{\mathcal{S}^a\}$ ;    // saturated goal acceleration (2.15)
    /* ===== advance current state (2.6) ===== */
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{f}^v(\boldsymbol{v}_k, \boldsymbol{a}_k^{\text{sat}}, \Delta t)$;
    $\boldsymbol{r}_{k+1} \leftarrow \boldsymbol{f}^r(\boldsymbol{r}_k, \boldsymbol{v}_k, \boldsymbol{a}_k^{\text{sat}}, \Delta t)$;
```

Let $\boldsymbol{R} \in \mathrm{SO}(3)$ be a rotation matrix that describes the rotation of an orthonormal basis from a fixed inertial frame to the body fixed frame. The angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ describes the angular velocity between the inertial frame and the body fixed frame w.r.t. the inertial frame. To constrain the angular speed $\|\boldsymbol{\omega}\|_2$ in the same fashion as the translational case, we start from the relation between the rotation matrix $\boldsymbol{R}$ and angular velocity $\boldsymbol{\omega}$. It is given by the vector cross product

$$\dot{\boldsymbol{R}} = \boldsymbol{\omega} \times \boldsymbol{R} \tag{2.18}$$

and can be expressed in matrix form

$$\dot{\boldsymbol{R}} = \underbrace{\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}}_{=:[\boldsymbol{\omega}]_\times} \boldsymbol{R} \tag{2.19}$$

where $[\,\cdot\,]_\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ denotes the skew operator. Note that $[\boldsymbol{\omega}]_\times$ is also called the SO(3) associated Lie algebra $\mathfrak{so}3$. We also use the inverse operator $[\,\cdot\,]_\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ that recovers the vector components of a skew symmetric matrix and define it as

$$\Big[[\boldsymbol{\omega}]_\times\Big]_\vee := \boldsymbol{\omega} \qquad \text{where } \boldsymbol{\omega} \in \mathbb{R}^3, [\boldsymbol{\omega}]_\times \in \mathfrak{so}3. \tag{2.20}$$

While (2.19) suggests an exponential integration of the form

$$\boldsymbol{R}(t+T) = \exp\left(\int\limits_t^{t+T} [\boldsymbol{\omega}(\tau)]_\times \,\mathrm{d}\tau\right) \boldsymbol{R}(t), \tag{2.21}$$

this is only leads to correct orientations in the case of a fixed rotation axis $\boldsymbol{\omega}/\|\boldsymbol{\omega}\|_2$, due to the definition of the matrix exponential as a power series and the non-commutativity of the elements in SO(3). This leads to various iterative integration schemes, where the axis is assumed to be fixed only for a single small time step $\varDelta t$. See [Boy17] for a detailed derivation and comparison of different schemes.

If (2.21) is solved for small time steps T, the error that comes from non-commutativity of SO(3) is negligible and thus often used for iterative schemes with small sampling times $\varDelta t$. However, the crucial step of finding the new goal velocity $\boldsymbol{v}_{k+1}^{\text{goal}}$ from (2.11) is accomplished by solving the time integration over a time span of $T_{\min}^{\text{tra}}$, which cannot be assumed to be small.

The chosen approach in this chapter is the use of the Magnus expansion as originally proposed in [Mag54]. This method provides an analytical framework for finding the solution of linear, time-variant matrix differential equations such as (2.19).

2.3.2.1 Magnus Expansion

Opposed to the iterative integration methods mentioned above, the Magnus expansion gives an exact solution, important for longer time ranges. The reason that (2.21) is not correct for the general rotational case lies in the non-commutativity of matrices $[\boldsymbol{\omega}]_\times$. The non-vanishing portion of the commutation is denoted as the commutator

$$[A, B] := AB - BA, \quad A \in \mathfrak{g}, B \in \mathfrak{g} \tag{2.22}$$

and naturally arises in Lie structures such as the Lie algebra $\mathfrak{g}$. Magnus' treatment is given in a more general problem, however our explanation refers to a real-valued time-variant matrix differential equation of the form (2.19).

Magnus' proposal is to find a solution for (2.19) in the form of a true matrix exponential

$$\boldsymbol{R}(t+T) = \exp\left([\boldsymbol{\varOmega}(t,T)]_\times\right) \boldsymbol{R}(t) \tag{2.23}$$

with a series expansion

$$[\boldsymbol{\varOmega}(t,T)]_\times := \sum_{i=1}^{\infty} [\boldsymbol{\varOmega}_i(t,T)]_\times \tag{2.24}$$

that is referred to as Magnus expansion. It is an infinite power series, however as pointed out in [BCOR09] it is usually sufficiently accurate for applications to truncate the series after the first three terms which are given as

$$[\boldsymbol{\varOmega}_1(t,T)]_\times := \int_t^{t+T} [\boldsymbol{\omega}(t_1)]_\times \,\mathrm{d}t_1 \tag{2.25}$$

$$[\boldsymbol{\varOmega}_2(t,T)]_\times := \frac{1}{2}\int_t^{t+T} \left[[\boldsymbol{\omega}(t_1)]_\times , \int_t^{t_1} [\boldsymbol{\omega}(t_2)]_\times \,\mathrm{d}t_2 \right] \mathrm{d}t_1 \tag{2.26}$$

$$[\boldsymbol{\varOmega}_3(t,T)]_\times := \frac{1}{6}\int_t^{t+T} \left[[\boldsymbol{\omega}(t_1)]_\times , \int_t^{t_1} \left[[\boldsymbol{\omega}(t_2)]_\times , \int_t^{t_2} [\boldsymbol{\omega}(t_3)]_\times \,\mathrm{d}t_3 \right] \mathrm{d}t_2 \right] \mathrm{d}t_1 \tag{2.27}$$

$$+ \frac{1}{6}\int_t^{t+T} \left[\left[[\boldsymbol{\omega}(t_1)]_\times , \int_t^{t_1} [\boldsymbol{\omega}(t_2)]_\times \,\mathrm{d}t_2 \right] , \int_t^{t_2} [\boldsymbol{\omega}(t_3)]_\times \,\mathrm{d}t_3 \right] \mathrm{d}t_1. \tag{2.28}$$

Note that all brackets used refer to the commutator defined in (2.22). For an explicit solution of the fourth order term as well as a recursive scheme to calculate Magnus expansion terms of arbitrary order we refer to [BCOR09].

The first three terms of the Magnus expansion (2.24) can be calculated explicitly for our system (2.19), and read

$$\boldsymbol{\Omega}_1(t_k + T) := \boldsymbol{\omega}_k T + \boldsymbol{\alpha}_k \frac{T^2}{2} \tag{2.29}$$

$$\boldsymbol{\Omega}_2(t_k + T) := [\boldsymbol{\alpha}_k]_\times \, \boldsymbol{\omega}_k \frac{T^3}{12} \tag{2.30}$$

$$\boldsymbol{\Omega}_3(t_k + T) := [\boldsymbol{\alpha}_k]_\times \, [\boldsymbol{\alpha}_k]_\times \, \boldsymbol{\omega}_k \frac{T^5}{240}. \tag{2.31}$$

The operator $[\cdot]_\times$ again denotes the skew symmetric matrix operator defined in (2.19). The Magnus expansion can thus be calculated as

$$\boldsymbol{\Omega}(t_k + T) := \boldsymbol{\Omega}_1(t_k + T) + \boldsymbol{\Omega}_2(t_k + T) + \boldsymbol{\Omega}_3(t_k + T) \tag{2.32a}$$

$$= \boldsymbol{M}(\boldsymbol{\alpha}_k, T)\boldsymbol{\omega}_k + \boldsymbol{\alpha}_k \frac{T^2}{2} \tag{2.32b}$$

with

$$\boldsymbol{M}(\boldsymbol{\alpha}_k, T) := \boldsymbol{I}_{3\times 3} T + [\boldsymbol{\alpha}_k]_\times \frac{T^3}{12} + [\boldsymbol{\alpha}_k]_\times \, [\boldsymbol{\alpha}_k]_\times \frac{T^5}{240} \tag{2.33}$$

where $\boldsymbol{I}_{3\times 3}$ is the identity matrix.

2.3.2.2 OTG for SO(3) using Rotation Matrices

The system dynamics of the rotational portion can now be stated as

$$\boldsymbol{\omega}(t_k + T) = \boldsymbol{f}^{\omega}(\boldsymbol{\omega}_k, \boldsymbol{\alpha}_k, T) := \boldsymbol{\omega}_k + \boldsymbol{\alpha}_k T \tag{2.34a}$$

$$\boldsymbol{R}(t_k + T) = \boldsymbol{f}^{R}(\boldsymbol{R}_k, \boldsymbol{\omega}_k, \boldsymbol{\alpha}_k, T) := \exp\left(\left[\boldsymbol{M}(\boldsymbol{\alpha}_k, T)\boldsymbol{\omega}_k + \boldsymbol{\alpha}_k \frac{T^2}{2}\right]_\times\right) \boldsymbol{R}_k \tag{2.34b}$$

where we see that unlike in the translation mapping $\boldsymbol{f}^r$, the orientation mapping $\boldsymbol{f}^R$ is *R5* and strongly coupled.

Analog to the algorithm for translation, the minimum time needed to drive the rotational velocity error $\boldsymbol{e}_k^{\omega} := \boldsymbol{\omega}_k^{\text{des}} - \boldsymbol{\omega}_k$ to zero is

$$\bar{T}_{\text{min}}^{\text{rot}} := \frac{\|\boldsymbol{e}_k^{\omega}\|_2}{\alpha_{\text{max}}}. \tag{2.35}$$

Discretizing the value w.r.t. the given sampling time $\varDelta t$ reads

$$T_{\text{min}}^{\text{rot}} := \max\left\{\left\lceil \frac{\bar{T}_{\text{min}}^{\text{rot}}}{\varDelta t} \right\rceil, 1\right\} \varDelta t. \tag{2.36}$$

The corresponding angular acceleration is

$$\bar{\boldsymbol{\alpha}}_k := \frac{\boldsymbol{e}_k^{\omega}}{T_{\min}^{\text{rot}}} \tag{2.37}$$

and is analog to $\boldsymbol{v}_k^{\text{goal}}$ from (2.10). It is used to find the new goal velocity $\boldsymbol{\omega}_k^{\text{goal}}$ by solving $\boldsymbol{r}_k^{\text{des}} = \boldsymbol{f}^R(\boldsymbol{R}_k, \boldsymbol{\omega}_k, \bar{\boldsymbol{\alpha}}_k, T_{\min}^{\text{rot}})$ for

$$\boldsymbol{\omega}_k^{\text{goal}} := \boldsymbol{M}(\bar{\boldsymbol{\alpha}}_k, T_{\min}^{\text{rot}})^{-1} \left(\left[\log(\boldsymbol{R}^{\text{des}} {\boldsymbol{R}_k}^{\top}) \right]_{\vee} - \bar{\boldsymbol{\alpha}}_k \frac{(T_{\min}^{\text{rot}})^2}{2} \right), \tag{2.38}$$

which in comparison to our treatment with Euler angle representation in [HGW17] is completely singularity-free and unambiguous. The proof that $\boldsymbol{M}$ is invertible for any $t \neq 0$, is provided in the appendix.

The goal velocity for the next time step is again found by advancing the previous equation for a single iteration

$$\begin{aligned} \boldsymbol{\omega}_{k+1}^{\text{goal}} &:= \boldsymbol{f}^{\omega}(\boldsymbol{\omega}_k^{\text{goal}}, \bar{\boldsymbol{\alpha}}_k, \Delta t) \qquad (2.39) \\ &= \boldsymbol{M}(\bar{\boldsymbol{\alpha}}_k, T_{\min}^{\text{rot}})^{-1} \left(\left[\log(\boldsymbol{R}^{\text{des}} {\boldsymbol{R}_k}^{\top}) \right]_{\vee} - \bar{\boldsymbol{\alpha}}_k \frac{(T_{\min}^{\text{rot}})^2}{2} \right) + \bar{\boldsymbol{\alpha}}_k \Delta t. \qquad (2.40) \end{aligned}$$

The constraints (2.4c) and (2.4d) are again incorporated by defining saturation factor sets $\mathcal{S}^{\omega} \subset \mathbb{R}^+$ and $\mathcal{S}^{\alpha} \subset \mathbb{R}^+$ for the angular velocity and acceleration respectively. The saturated rotation velocity is thus found by

$$\boldsymbol{\omega}_{k+1}^{\text{sat}} := \boldsymbol{\omega}_{k+1}^{\text{goal}} \min\{\mathcal{S}^{\omega}\}, \tag{2.41}$$

using the factorization set definition

$$\mathcal{S}^{\omega} := \{1, \sigma^{\omega}\} \quad \text{with} \quad \sigma^{\omega} := \frac{\omega_{\max}}{\left\| \boldsymbol{\omega}_{k+1}^{\text{goal}} \right\|_2}. \tag{2.42}$$

This again leads to the find goal acceleration

$$\boldsymbol{\alpha}_k^{\text{goal}} := \frac{\boldsymbol{\omega}_{k+1}^{\text{sat}} - \boldsymbol{\omega}_k}{\Delta t}. \tag{2.43}$$

Similarly, defining the acceleration saturation set

$$\mathcal{S}^{\alpha} := \{1, \sigma^{\alpha}\} \quad \text{with} \quad \sigma^{\alpha} := \frac{\alpha_{\max}}{\left\| \boldsymbol{\alpha}_k^{\text{goal}} \right\|_2} \tag{2.44}$$

and applying it to the goal acceleration

$$\boldsymbol{\alpha}_k^{\text{sat}} := \boldsymbol{\alpha}_k^{\text{goal}} \min\{\mathcal{S}^{\alpha}\} \tag{2.45}$$

results in the saturated angular acceleration $\boldsymbol{\alpha}_k^{\text{sat}}$. This acceleration vector is eventually applied to the SO(3) system dynamics (2.34), to advance the current orientation states

$$\begin{aligned} \boldsymbol{\omega}_{k+1} &= \boldsymbol{f}^{\omega}(\boldsymbol{\omega}_k, \boldsymbol{\alpha}_k^{\text{sat}}, \Delta t) \\ \boldsymbol{R}_{k+1} &= \boldsymbol{f}^{R}(\boldsymbol{R}_k, \boldsymbol{\omega}_k, \boldsymbol{\alpha}_k^{\text{sat}}, \Delta t) \end{aligned} \tag{2.46}$$

which concludes the algorithm using rotation matrices.

2.3.2.3 OTG for SO(3) using Unit Quaternions

The above algorithm can be straight forward translated to unit quaternion representation of SO(3). This reduces computation time, as discussed in Section 2.5. Let $\mathfrak{q} \in \mathbb{H}$ describe the 3D orientation as a unit quaternion, i.e. $\|\mathfrak{q}\|_2 = 1$. If we consider a quaternion $\mathfrak{q} := (\mathfrak{q}_w, \mathfrak{q}_v)$ consisting of a scalar part $\mathfrak{q}_w$ and a vector part $\mathfrak{q}_v$, the angular velocity $\boldsymbol{\omega}$ can be expressed as a pure quaternion $(0, \boldsymbol{\omega})$, i.e. with zero scalar part. We introduce the mappings

$$[\boldsymbol{\omega}]_{\mathfrak{q}} := (0, \boldsymbol{\omega}) \qquad \mathbb{R}^3 \to \mathbb{H} \tag{2.47}$$

$$[(0, \boldsymbol{\omega})]_{\mathfrak{q}^{-1}} := \boldsymbol{\omega} \qquad \mathbb{H} \to \mathbb{R}^3 \tag{2.48}$$

to specify notation. With this notation, the differential equation relating time derivatives of $\mathfrak{q}$ with the angular velocity $\boldsymbol{\omega}$ reads

$$\dot{\mathfrak{q}}(t) = \frac{1}{2} [\boldsymbol{\omega}(t)]_{\mathfrak{q}} \, \mathfrak{q}(t). \tag{2.49}$$

Magnus' idea to solve the differential equation results in solving (2.49) in the form

$$\mathfrak{q}(t+T) = \exp\left(\frac{1}{2} [\boldsymbol{\Omega}(t, T)]_{\mathfrak{q}}\right) \mathfrak{q}(t). \tag{2.50}$$

Note that unlike in the Matrix version of Magnus' proposal from (2.23), $\boldsymbol{\Omega}$ does not occur with the skew symmetric matrix operator $[\,\cdot\,]_\times$. The Magnus expansion of $\boldsymbol{\Omega}$ and its approximation (2.32) however, remain the same.

The mapping that advance the current quaternion in time, hence is

$$\mathfrak{q}(t+T) = \boldsymbol{f}^{\mathfrak{q}}(\mathfrak{q}_k, \boldsymbol{\omega}_k, \boldsymbol{\alpha}_k, T) := \exp\left(\frac{1}{2}\left[\boldsymbol{M}(\boldsymbol{\alpha}_k, T)\boldsymbol{\omega}_k + \boldsymbol{\alpha}_k \frac{T^2}{2}\right]_{\mathfrak{q}}\right) \mathfrak{q}(t). \tag{2.51}$$

The definition of $\boldsymbol{M}$ in (2.33) as well as velocity progression (2.34a) are unchanged. Therefore, the the only adjusted equations for the algorithm outlined in Section 2.3.2 are the definition of $\boldsymbol{\omega}_k^{\text{goal}}$ (2.38) w.r.t. quaternions

$$\boldsymbol{\omega}_k^{\text{goal}} = 2\boldsymbol{M}(\bar{\boldsymbol{\alpha}}_k, T_{\text{min}}^{\text{rot}})^{-1}\left(\left[\log(\mathfrak{q}^{\text{des}}\mathfrak{q}_k{}^{-1})\right]_{\mathfrak{q}^{-1}} - \boldsymbol{\alpha}_k \frac{(T_{\text{min}}^{\text{rot}})^2}{2}\right) \tag{2.52}$$

and consequently

$$\boldsymbol{\omega}_{k+1}^{\text{goal}} = 2\boldsymbol{M}(\bar{\boldsymbol{\alpha}}_k, T_{\text{min}}^{\text{rot}})^{-1}\left(\left[\log(\mathfrak{q}^{\text{des}}\mathfrak{q}_k{}^{-1})\right]_{\mathfrak{q}^{-1}} - \bar{\boldsymbol{\alpha}}_k \frac{(T_{\text{min}}^{\text{rot}})^2}{2}\right) + \bar{\boldsymbol{\alpha}}_k \Delta t. \tag{2.53}$$

The algorithm for OTG on rotations using Quaternions is listed in Algorithm 2.2.

2.3.3 Synchronization of Translation and Rotation

While $T_{\text{min}}^{\text{tra}}$ and $T_{\text{min}}^{\text{rot}}$ from (2.7) and (2.36) assure time synchronization within the translational and rotational DOFs resp., the two groups are not yet synchronized with each other. Complete

Algorithm 2.2: OTG on SO(3) with magnitude constraints using Quaternions

```
Function OTGquaternions
    Data   : current state ($\mathfrak{q}_k, \boldsymbol{\omega}_k$), desired state ($\mathfrak{q}_k$des, $\boldsymbol{\omega}_k^{\text{des}}$),
             constraints ($\omega_{\max}, \alpha_{\max}$), sampling time $\Delta t$
    Result : $\mathfrak{q}_{k+1}, \boldsymbol{\omega}_{k+1}$
    /* ===== find new goal velocity ===== */
    $T_{\text{min}}^{\text{rot}} \leftarrow \max\left\{\left\lceil \frac{\|\boldsymbol{\omega}_k^{\text{des}} - \boldsymbol{\omega}_k\|_2}{\alpha_{\max}} \right\rceil, 1\right\} \Delta t$ ;    // discretized minimum Time (2.36)
    $\bar{\boldsymbol{\alpha}}_k \leftarrow \frac{\boldsymbol{\omega}_k^{\text{des}} - \boldsymbol{\omega}_k}{T_{\text{min}}^{\text{rot}}}$ ;    //  corresponding acceleration (2.37)
    $\boldsymbol{M} \leftarrow \boldsymbol{I}_{3\times3} T_{\text{min}}^{\text{rot}} + [\bar{\boldsymbol{\alpha}}_k]_\times \frac{\left(T_{\text{min}}^{\text{rot}}\right)^3}{12} + [\bar{\boldsymbol{\alpha}}_k]_\times [\bar{\boldsymbol{\alpha}}_k]_\times \frac{\left(T_{\text{min}}^{\text{rot}}\right)^5}{240}$ ;    // Magnus Series (2.33)
    $\boldsymbol{\omega}_{k+1}^{\text{goal}} \leftarrow 2\boldsymbol{M}^{-1} \left( \left[\log(\mathfrak{q}^{\text{des}} \mathfrak{q}_k{}^{-1})\right]_{\mathfrak{q}^{-1}} - \bar{\boldsymbol{\alpha}}_k \frac{\left(T_{\text{min}}^{\text{rot}}\right)^2}{2} \right) + \bar{\boldsymbol{\alpha}}_k \Delta t$ ;    // goal velocity (2.53)
    $\mathcal{S}^\omega \leftarrow$ collectVelocitySaturationFactors($\boldsymbol{\omega}_{k+1}^{\text{goal}}$,$\omega_{\max}$) ;
    $\boldsymbol{\omega}_{k+1}^{\text{sat}} \leftarrow \boldsymbol{\omega}_{k+1}^{\text{goal}} \min\{\mathcal{S}^\omega\}$ ;    // saturate goal velocity
    /* ===== find goal acceleration ===== */
    $\boldsymbol{\alpha}_k^{\text{goal}} \leftarrow \frac{\boldsymbol{\omega}_{k+1}^{\text{sat}} - \boldsymbol{\omega}_k}{\Delta t}$ ;    // goal acceleration (2.43)
    $\mathcal{S}^\alpha \leftarrow$ collectVelocitySaturationFactors($\boldsymbol{\alpha}_k^{\text{goal}}$,$\alpha_{\max}$) ;
    $\boldsymbol{\alpha}_k^{\text{sat}} \leftarrow \boldsymbol{\alpha}_k^{\text{goal}} \min\{\mathcal{S}^\alpha\}$ ;    // saturate goal acceleration (2.45)
    /* ===== advance current state (2.34a) and (2.51) ===== */
    $\boldsymbol{\omega}_{k+1} \leftarrow \boldsymbol{f}^\omega(\boldsymbol{\omega}_k, \boldsymbol{\alpha}_k^{\text{sat}}, \Delta t)$;
    $\mathfrak{q}_{k+1} \leftarrow \boldsymbol{f}^\alpha(\mathfrak{q}_k, \boldsymbol{\omega}_k, \boldsymbol{\alpha}_k^{\text{sat}}, \Delta t)$;
```

synchronization of the whole 6D movement can be achieved by coupling the translation and rotation at two distinct places in the algorithm. First, the more restrictive minimum time

$$T_{\text{min}} := \max\left(T_{\text{min}}^{\text{tra}}, T_{\text{min}}^{\text{rot}}\right) \tag{2.54}$$

is used for finding both acceleration vectors $\boldsymbol{a}_k$ and $\boldsymbol{\alpha}_k$ in (2.9) and (2.37). Second, the saturation of the goal velocities are matched by combining their saturation factor sets (2.12) and (2.42) as

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \mathcal{S}^\omega \tag{2.55a}$$

$$\mathcal{S}^\omega \leftarrow \mathcal{S}^v \cup \mathcal{S}^\omega \tag{2.55b}$$

An example of this synchronization is shown in Figure 2.2.

2.4 Extensions for HRC Scenarios

While the presented algorithm considers a free 6D motion of e.g. the tool center point (TCP) of a robot, it does not depend on any specific robot kinematics In the following Section, we present some interesting constraint extensions particularly relevant in HRC context. The examples highlight the flexibility of our approach and demonstrate the straight forward integration for additional requirements by adding appropriate factors to the saturation sets $\mathcal{S}$. The factors are found via a 3-step process:

Step 1 formulate a scalar inequality condition e.g.

$$\|\boldsymbol{v}\|_2 \leq v_{\max}$$

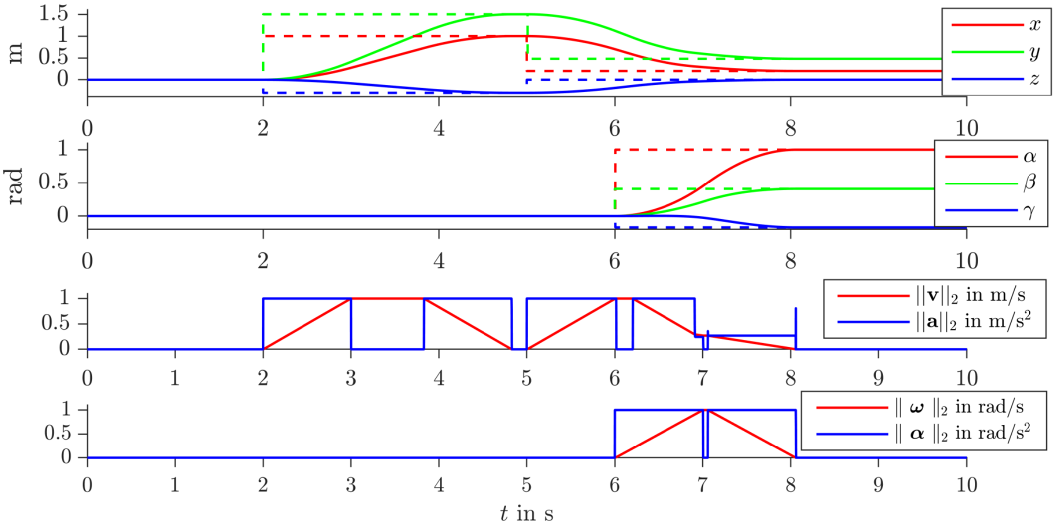


Figure 2.2: Time synchronization of translation and rotation. Dashed lines are set-points and solid lines are the actual values. At t =6 s the jump in desired rotation causes an adjustment of the deceleration phase of the translation trajectories in order to synchronize the movements. Note that absolute speed and acceleration are plotted as magnitude values.

Step 2 derive the normalization factor σ that yields equality to 1 e.g.

$$\|\boldsymbol{v}\|_2\,\sigma = 1 \quad \text{where} \quad \sigma = \frac{v_{\max}}{\|\boldsymbol{v}\|_2}$$

Step 3 add the factor to the corresponding saturation factor set e.g.

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma\}$$

Note, that in the following all constraints refer to the variables at the next discrete time instance $k+1$. Therefore we suppress the discrete time index $k+1$ in the remainder of this section for better readability. Further, Sections 2.4.2 to 2.4.4 require that translation and rotation are synchronized as outlined in Section 2.3.3.

2.4.1 Direction-specific constraints

In robot applications it is often very useful to constrain different directions of movement independently. While the OTGs mentioned in the related work of Section 2.1.1 offer the possibility to constrain certain axes of the underlying inertial coordinate system directly, our framework can be easily extended to the more general case of incorporating constraints in arbitrary directions. Possible scenarios in HRC that require these type of constraints, are e.g. the dynamic limitation of robot movements towards the human or obstacles. The requirement of constraining the velocity $\boldsymbol{v}^{\text{tra}}$ along a given direction vector $\boldsymbol{r}^{\text{dir}} \in \mathbb{R}^3$ is formulated as

$$(\boldsymbol{v}^{\text{tra}})^\top \frac{\boldsymbol{r}^{\text{dir}}}{\|\boldsymbol{r}^{\text{dir}}\|_2} \leq v_{\max}, \tag{2.56}$$

where the $\boldsymbol{v}^{\mathrm{tra}}$ is projected onto the direction vector $\boldsymbol{r}^{\mathrm{dir}}$. The necessary saturation factor that complies with this constraint is given as

$$\sigma^{\mathrm{dir}} := \frac{v_{\max} \left\| \boldsymbol{r}^{\mathrm{dir}} \right\|_2}{(\boldsymbol{v}^{\mathrm{tra}})^\top \boldsymbol{r}^{\mathrm{dir}}} \tag{2.57}$$

and adding it to the velocity saturation set

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma^{\mathrm{dir}}\} \tag{2.58}$$

incorporates this constraint in (2.13) of the algorithm. Note that this strategy does not only allow constraining orthogonal axes, but several arbitrary directions.

2.4.2 Combining Translation and Rotation Constraints

Whenever humans work in close proximity with robots, robot constraints should be defined as intuitive as possible. Besides constraining linear and angular velocity separately as done in Section 2.3, we outline a strategy to combine the two into a single intuitive constraint.

Constraining Total Velocity of Specific Points Instead of constraining the angular speed $\|\boldsymbol{\omega}\|_2 \leq \omega_{\max}$ of the TCP directly, it is sometimes more intuitive to constrain the total linear velocity

$$\left\| \boldsymbol{v}^{\mathrm{tot}}(\boldsymbol{r}) \right\|_2 := \left\| \boldsymbol{v}^{\mathrm{tra}} + \boldsymbol{v}^{\mathrm{rot}}(\boldsymbol{r}) \right\|_2 \leq v_{\max} \tag{2.59}$$

of a given vector $\boldsymbol{r}^{\in \mathbb{R}^3}$, relative to the TCP. The translational and rotational velocity at the TCP is denoted as $\boldsymbol{v}^{\mathrm{tra}} := \boldsymbol{v}$ and $\boldsymbol{v}^{\mathrm{rot}} := \boldsymbol{\omega} \times \boldsymbol{r}$ resp. This can be a specific point on the geometric robot-object compound that is fixed to the TCP, e.g. the robots wrist as illustrated in Figure 2.3. Note that all variables in (2.59) are given in a world-fixed inertial frame I. In case a specific point is given w.r.t. the end-effector coordinate system E, it needs to be transformed i.e.

$$_I\boldsymbol{r} = \boldsymbol{R}_{IE}\, {_E\boldsymbol{r}}, \tag{2.60}$$

where $\boldsymbol{R}_{IE}$ rotates the vector from E to the inertial frame I. Let θ denote the angle between $\boldsymbol{\omega}$ and $\boldsymbol{r}$. The fact that the magnitude

$$\left\| \boldsymbol{v}^{\mathrm{rot}}(\boldsymbol{r}) \right\|_2 = \|\boldsymbol{\omega} \times \boldsymbol{r}\|_2 = \|\boldsymbol{\omega}\|_2 \|\boldsymbol{r}\|_2 \sin(\theta) \tag{2.61}$$

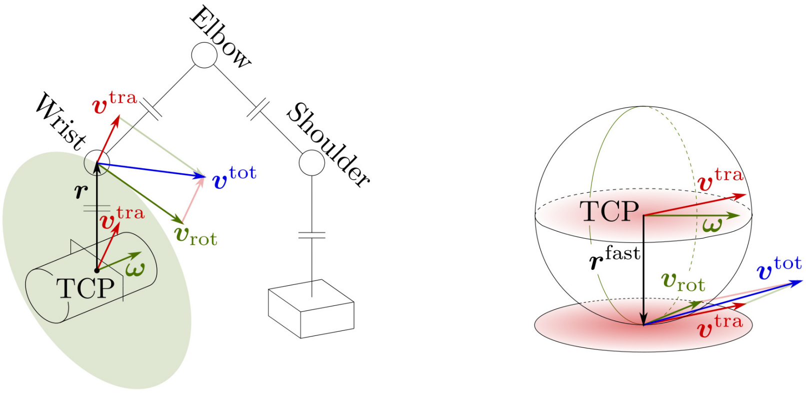
is linear in $\boldsymbol{\omega}$, allows us to saturate $\|\boldsymbol{v}^{\mathrm{tot}}\|_2 \leq v_{\max}$ by calculating a common factor

$$\sigma^{\mathrm{tot}}(\boldsymbol{r}) := \frac{v_{\max}}{\|\boldsymbol{v}^{\mathrm{tot}}(\boldsymbol{r})\|_2} \tag{2.62}$$

and adding it to the factorization set

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma^{\mathrm{tot}}\} \tag{2.63}$$

If n multiple points are to be considered e.g. to outline a convex polytope around the TCP, multiple factors $\sigma^{;i}_{\mathrm{tot}}$ for $i = [1, n]$ can be calculated and added to the factorization sets in the same fashion. Note that for the TCP itself, $\boldsymbol{r}$ is zero by definition and thus $\boldsymbol{v}^{\mathrm{tot}} = \boldsymbol{v}^{\mathrm{tra}}$, which is already considered in the basic algorithm (2.13).


Elbow
Wrist
Shoulder
$\boldsymbol{v}^{\text{tra}}$
$\boldsymbol{v}^{\text{tot}}$
$\boldsymbol{r}$
$\boldsymbol{v}_{\text{rot}}$
$\boldsymbol{\omega}$
TCP
$\boldsymbol{r}^{\text{fast}}$

(a) Specific point

(b) Safety sphere

Figure 2.3: Illustration of (a) total velocity of e.g. the robot wrist at a fixed vector $\boldsymbol{r}$ and (b) projection of the angular velocity for identifying the fastest point $\boldsymbol{r}^{\text{fast}}$ within a user-defined safety sphere of radius r

Constraining Total Velocity within Safety Sphere Instead of defining specific points around the TCP, it also possible to define a safety sphere $\mathbb{S} := \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r\}$ with radius r around the TCP. It is then desired that no point within the sphere is allowed to exceed the speed limit $v_{\max}$. This extends the before mentioned strategy to first identifying the fastest point $\boldsymbol{r}^{\text{fast}}$ within the sphere.

To maximize (2.61), $\boldsymbol{r}$ must *a)* lie on the sphere surface i.e. have length r and *b)* be perpendicular to the rotation vector $\boldsymbol{\omega}$. Further, to maximize $\boldsymbol{v}^{\text{tot}}$, the vector $\boldsymbol{r}$ has to be chosen such that *c)* the cross product $\boldsymbol{\omega} \times \boldsymbol{r}$ lies in the plane spanned by $\boldsymbol{\omega}$ and $\boldsymbol{v}^{\text{tra}}$, and *d)* the angle between $\boldsymbol{v}^{\text{tra}}$ and $\boldsymbol{v}^{\text{rot}}$ is below 90deg. See Figure 2.3 for an illustration.

Thus, the fastest point $\boldsymbol{r}^{\text{fast}} \in \mathbb{S}$ is given as

$$\boldsymbol{r}^{\text{fast}} := \begin{cases} -\dfrac{\boldsymbol{\omega} \times \boldsymbol{v}^{\text{tra}}}{\|\boldsymbol{\omega} \times \boldsymbol{v}^{\text{tra}}\|_2} r & \text{for } \boldsymbol{\omega} \times \boldsymbol{v}^{\text{tra}} \neq \boldsymbol{0} \\ \dfrac{\boldsymbol{\omega} \times \boldsymbol{r}^{\perp}}{\|\boldsymbol{\omega} \times \boldsymbol{r}^{\perp}\|_2} r & \text{else} \end{cases} \tag{2.64}$$

where $\boldsymbol{r}^{\perp} \in \{\boldsymbol{x} \in \mathbb{R}^3 \backslash \{\boldsymbol{0}\} \,|\, \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle = 0\}$ is an arbitrary vector perpendicular to $\boldsymbol{\omega}$. Its total speed is

$$\left\|\boldsymbol{v}^{\text{tot}}(\boldsymbol{r}^{\text{fast}})\right\|_2 = \left\|\boldsymbol{v}^{\text{tra}} + \boldsymbol{\omega} \times \boldsymbol{r}^{\text{fast}}\right\|_2 \tag{2.65}$$

which leads to the saturation factor

$$\sigma^{\text{sphere}} := \frac{v_{\max}}{\|\boldsymbol{v}^{\text{tot}}(\boldsymbol{r}^{\text{fast}})\|_2}. \tag{2.66}$$

Figure 2.4 shows the effect of this safety sphere constraint, if added to the set

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma^{\text{sphere}}\} \tag{2.67}$$

and applied in the algorithm.

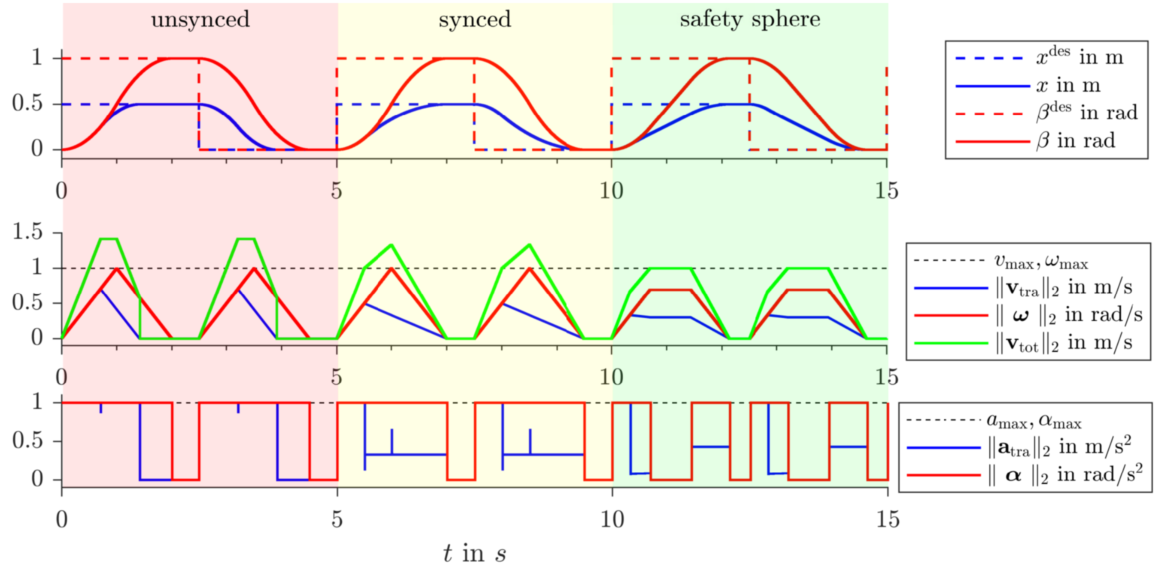


Figure 2.4: Combination of translational an angular velocity constraints, by defining a safety sphere of radius $r = 1\,\mathrm{m}$ around the TCP. For demonstration purposes the PTP set-points are chosen in single axes directions x and rotation around y by an angle β for translation and rotation respectively. Three different modes are shown, applying norm constraints $v_{\max} = \omega_{\max} = 1$. The first motion 1 s to 5 s is without synchronization. The second motion 5 s to 10 s is time synchronized according to Section 2.3.3, and the third motion 11 s to 15 s is time synchronized while additionally applying the safety sphere constraint from Section 2.4.2. Note that the change in the slope of $\|\boldsymbol{v}^{\text{tot}}\|_2$ at 10.3 s stems from the deceleration of $\boldsymbol{v}^{\text{tra}}$ to cruising speed. Both phases of the movement last 0.7 s and are dictated by $a_{\max}$ during deceleration.

2.4.3 Robot Joint Constraints

While the presented OTG algorithm can be directly used to generate 6D trajectories of e.g. the robot end-effector, joint velocity limits were not yet considered, as they depend on the robot-specific kinematic structure as well as the inverse kinematics solver in use. Because kinematic relations are usually highly *R5* due to rotary joints, the inverse kinematic solvers often make use of the linear relation

$$\dot{\boldsymbol{z}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.68}$$

between task space velocities e.g. $\dot{\boldsymbol{z}} = \left[\dot{\boldsymbol{v}}^\top, \dot{\boldsymbol{\omega}}^\top\right]^\top \in \mathbb{R}^6$ and joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$, via the Jacobian $\boldsymbol{J} = \frac{\partial \dot{\boldsymbol{z}}}{\partial \dot{\boldsymbol{q}}}$. If the expression $\dot{\boldsymbol{q}} = \mathrm{IK}(\dot{\boldsymbol{z}})$ of the inverse kinematics solver in use is known, the joint velocity limits can directly be considered in the algorithm. For demonstration purposes we assume that a damped pseudo-inverse solver of the form

$$\dot{\boldsymbol{q}} = \mathrm{IK}(\dot{\boldsymbol{z}}) := \underbrace{\boldsymbol{J}^\top \left(\boldsymbol{J}\boldsymbol{J}^\top + \alpha^2 \boldsymbol{I}_{n\times n}\right)^{-1}}_{=:\boldsymbol{J}^\dagger} \dot{\boldsymbol{z}} \tag{2.69}$$

is used. The identity matrix $\boldsymbol{I}_{n\times n}$ together with the parameter α introduces a damping effect on the inverse kinematics solution. This avoids singularity issues and is discussed in detail in [Bus04]. The individual constraints on the n joint velocities

$$-\dot{q}_i^{\max} \leq \dot{q}_i \leq \dot{q}_i^{\max} \quad \text{with } i = [1, n], \tag{2.70}$$

can concisely be written as

$$\|\boldsymbol{L}\dot{\boldsymbol{q}}\|_\infty \leq 1, \tag{2.71}$$

with the diagonal limit matrix $\boldsymbol{L} = \mathrm{diag}\left(\dot{q}_1^{\max}, \dots, \dot{q}_n^{\max}\right)^{-1}$. It contains reciprocal values of the joint-specific velocity limits of all n joints in the serial robot kinematics. The linearity of the velocity mapping (2.68) again admits a scaling factor

$$\sigma^{\dot{q}} := \|\boldsymbol{L}\dot{\boldsymbol{q}}\|_\infty^{-1} \tag{2.72a}$$

$$= \|\boldsymbol{L} \underbrace{\boldsymbol{J}^\dagger \dot{\boldsymbol{z}}}_{\mathrm{IK}(\dot{\boldsymbol{z}})} \|_\infty^{-1}, \tag{2.72b}$$

such that factorizing the joint velocities $\dot{\boldsymbol{q}}$ is equivalent to factorizing the task space velocities

$$\sigma^{\dot{q}} \dot{\boldsymbol{z}} = \sigma^{\dot{q}} \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.73}$$

directly. Thus, adding $\sigma^{\dot{q}}$ to the saturation factor set

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma^{\dot{q}}\} \tag{2.74}$$

constrains the joint velocities accordingly.

Note, that constraining the joint acceleration $\ddot{\boldsymbol{q}}$ in the same manner is not possible, because the kinematic relation for accelerations

$$\ddot{\boldsymbol{z}} = \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} + \boldsymbol{J}\ddot{\boldsymbol{q}} \tag{2.75a}$$

$$= \dot{\boldsymbol{J}}\boldsymbol{J}^\dagger \dot{\boldsymbol{z}} + \boldsymbol{J}\ddot{\boldsymbol{q}} \tag{2.75b}$$

couples task space velocity $\dot{\boldsymbol{z}}$ and acceleration $\ddot{\boldsymbol{z}}$. Thus, a linear relation such as (2.73), that elevates the joint constraint factor to task space, is not given.

2.4.4 Constraining Movement of the Whole Robot Kinematic

In a serial kinematic robot as usually used in a HRC context, the most important movements to constrain is the movement of the end-effector or TCP. However, there might be other points on the robot structure that reach even higher velocities. Thus, it is necessary to not only constrain the TCP movement itself, but extend the given speed constrain to every point in the robot kinematics structure in task space. Only then can official safety standards e.g. ISO/TS 15066 [ISO16] be satisfied. Given a conventional rigid link structure consisting of joint axes only orthogonal or parallel to the links, such as illustrated in Figure 2.3a, it is sufficient to check the set of axes intersecting points $\mathcal{I}$ of the kinematic for

$$\|\boldsymbol{v}_i\|_2 \leq v_{\max} \quad \forall i \in \mathcal{I}. \tag{2.76}$$

The velocities $\boldsymbol{v}_i$ are calculated by

$$\|\boldsymbol{v}_i\|_2 := \|\boldsymbol{J}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i\|_2 \tag{2.77}$$

using the corresponding Jacobian matrices $\boldsymbol{J}_i$ with the reduced joint velocity vectors $\dot{\boldsymbol{q}}$. In the illustrated 7DOF kinematic these are $\mathcal{I} = \{\text{shoulder, elbow, wrist, TCP}\}$. Every constraint leads to a saturation factor

$$\sigma_i^{\text{kin}} := \frac{v_{\max}}{\|\boldsymbol{v}_i\|_2} \quad \forall i \in \mathcal{I} \tag{2.78}$$

that as part of the sets

$$\mathcal{S}^v \leftarrow \mathcal{S}^v \cup \{\sigma_i^{\text{kin}}\} \tag{2.79}$$

slows down the end-effector velocity in the OTG whenever necessary.

Note that in the discussed conventional serial 7 DOF, this method is only relevant for constraining the elbow movement. The translational shoulder velocity is always zero and can thus be omitted. Further the TCP being the general point of interest, is by definition already constrained in the basic algorithm. The wrist can be considered independently of the kinematic structure in use, following the method in Section 2.4.2.

2.4.5 OTG Independent Constraints

Considering time variant velocity constraints opens up interesting application scenarios too. It allows for integrating OTG independent metrics to be considered in the OTG algorithm. An exemplary use-case in HRC, is a distance-sensitive velocity limitation. In terms of safety issues in HRC, the maximal allowed robot velocity in the proposed OTG is made dependent on the Euclidean distance d^{HR} between human and robot by using e.g. a Gaussian shaped weighting term

$$v_{\max}(d^{\text{HR}}) := \begin{cases} \exp\left(-\dfrac{(d^{\text{HR}} - d_{\min})^2}{2\nu^2}\right)\bar{v}_{\max} & \text{, for } d^{\text{HR}} \leq d_{\min} \\ \bar{v}_{\max} & \text{, else} \end{cases} \tag{2.80}$$

with the shaping factor ν. A shaping factor of $\frac{d_{\min}}{3}$ e.g. scales the velocity to $< 1\%$ for $d^{\text{HR}} \to 0$. Figure 2.5 shows the comparison between a fixed $v_{\max}$ and the distance sensitive implementation (2.80) for the case when the robot crosses 10 cm above the human hand. In this example it is further demonstrated how the OTG copes with measurement noise of the human position.

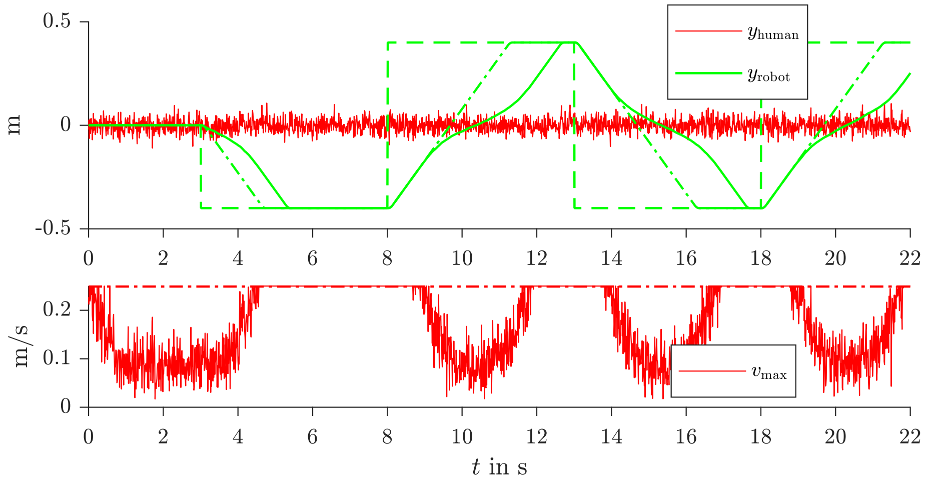


Figure 2.5: Hardware Experiment: distance sensitive $v_{\max}(d^{\text{HR}})$ with $\nu = \frac{d_{\min}}{3}$. Dashed lines are set-points, dash-dotted lines are fixed $v_{\max}$ and solid lines are $v_{\max}(d^{\text{HR}})$. The OTG clearly slows down whenever the distance becomes smaller $d_{\min} = 20\text{cm}$ and uses the user-defined constraint $v_{\max}$ otherwise.

2.5 Discussion

In this Section we will discuss the performance of the proposed algorithm in terms of transient behavior. We further give a runtime comparison to the state-of-the-art.

2.5.1 Transient behavior

Figure 2.6 shows the response of the approach to several step inputs and time-invariant constraints. Note, that for such inputs the algorithm results in a time-optimal bang-bang behavior in acceleration. The user constraints (2.4a) and (2.4b) are fulfilled at all times, while (2.7)-(2.9) assures time synchronization.

In case of trajectory following where the input trajectory fulfills the inequality constraints (2.4), the algorithm degenerates to

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k^{\text{des}} \tag{2.81a}$$

$$\mathfrak{q}_{k+1} = \mathfrak{q}_k^{\text{des}}. \tag{2.81b}$$

once the desired state is reached.

Thus, in our algorithm no explicit switching between different strategies is necessary. This means that the algorithm uses axes-synchronized acceleration bang-bang control in cases of constraint violating goal trajectories, no matter if they result from discontinuous set-point trajectories or merely from e.g. infeasible velocities. This reduces the error in position and velocity offset and eventually feeds through the target points whenever possible. An example of this behavior is shown in Figure 2.7.

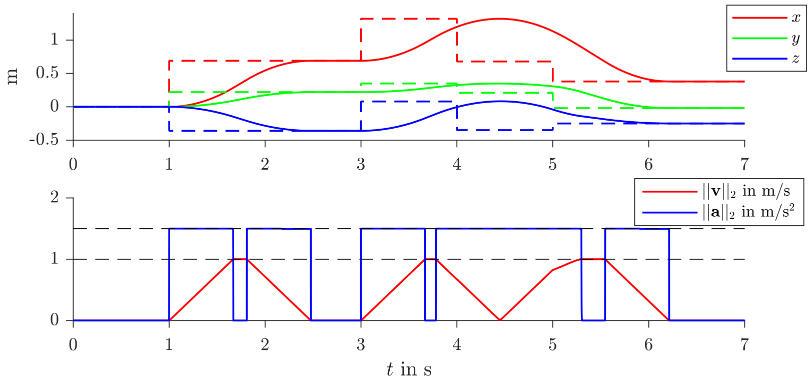


Figure 2.6: Trajectory generation for translation with hanging transition. Dashed lines are set-points/constraints and solid lines are the actual time synchronized values. The spikes in acceleration result from the transition phase due to time discretization. The change in set-point at $t = 5\,\mathrm{s}$ causes an adjustment in the direction of the acceleration vector, while its full magnitude $a_{\max} = 1.5\,\mathrm{m/s^2}$ is still exploited.

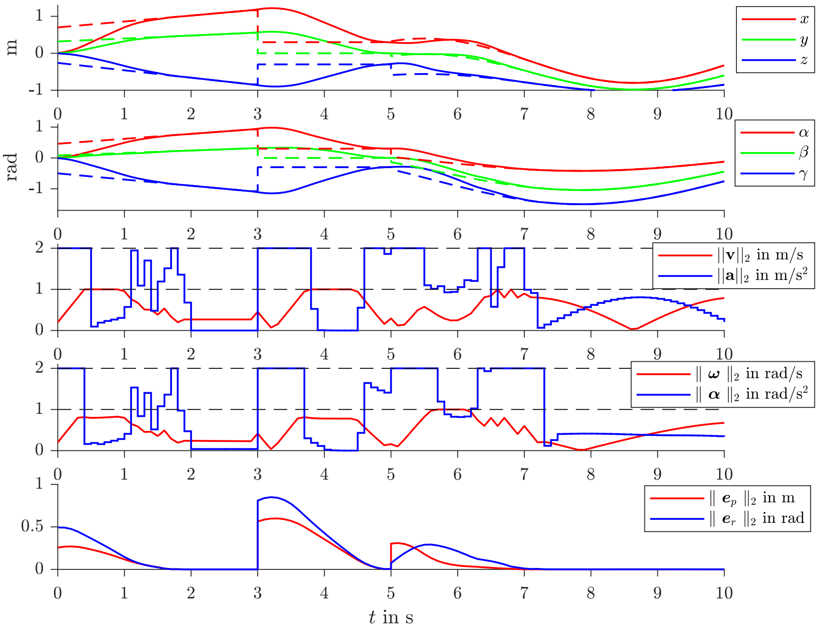


Figure 2.7: Time synchronous transition between acceleration bounded input trajectories and step inputs for a sample rate of 0.1 s. The discontinuous desired profile consists of three distinct segments. An initial ramp with offset (0 s to 3 s), followed by a constant (3 s to 6 s) and a sine with distinct frequencies (6 s to 10 s). State discrepancies at time $t = 0\,\mathrm{s}$ that usually result in an error state of the robot can be avoided. After the transition phase, the error for translation and rotation (bottom plot) converge to direct feedthrough of the desired trajectory as discussed in (2.81).

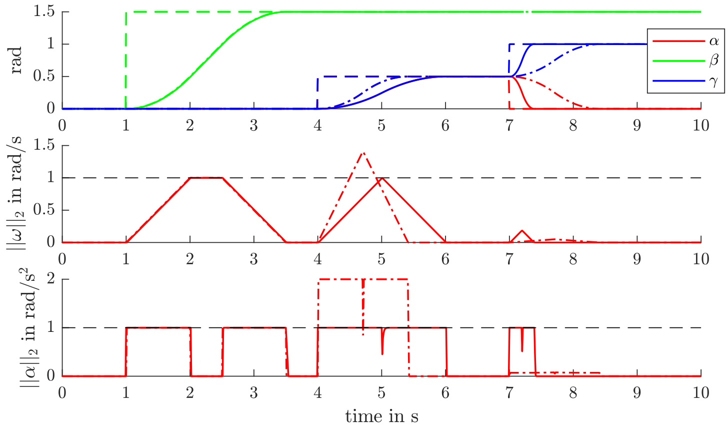


Figure 2.8: Comparison of our algorithm on SO(3) to the Reflexxes Motion Library, parametrizing Euler angles, in terms of magnitude velocity and acceleration. Note that in the given Euler angle sequence $X(\alpha) \to Y(\beta) \to Z(\gamma)$, the gimbal block singularity is reached at $\beta = \pi/2 \approx 1.57\,\mathrm{rad}$. Dashed lines are desired values/constraints, solid lines reflect our approach and dash-dotted lines result from the Reflexxes Motion Library. While α and γ accelerate in the same direction at $t = 4\,\mathrm{s}$ reaching almost double the constrained acceleration limit, constraining the Euler angle derivatives is over-restrictive at $t = 7\,\mathrm{s}$ due to the opposite movement directions.

2.5.2 Analytic vs. Geometric Angular Velocity Constraints

Current OTG algorithms mentioned in literature do either not provide synchronization treatment of the multi-dimensional case, or treat all DOFs as independent axes for trajectory generation. Directly applying joint related approaches to Euler angles for representing SO(3), typically are prone to so-called gimbal-lock-singularities. As analyzed in Section 2.3.2, instead of directly regarding constraints for the time derivatives of the Euler angles, we transfer the problem from this analytical subsequent velocity vector to the geometric angular velocities $\boldsymbol{\omega}$. Note that these two velocity vectors differ clearly in their physical interpretation. The time derivative of Euler angles describe consecutive angular velocities according to the given Euler sequence, whereas $\boldsymbol{\omega}$ contains simultaneous angular velocity around the base vectors at a given time instance. Constraining these two vectors directly, results obviously in different trajectories for a general case. In Figure 2.8 we demonstrate the divergence of the generated trajectories for some extreme cases. The closer the configuration gets to the singularity, the larger the discrepancy between Euler angle velocities and actual geometric angular velocities. The differences can be seen in velocity as well as acceleration level. We argue that constraining the angular velocity $\boldsymbol{\omega}$ in its magnitude gives raise to more intuitive trajectories than constraining the successive Euler angle velocities.

2.5.3 Runtime Analysis

As mentioned in the introduction, optimization based approaches especially when considering the required *R5* magnitude constraints as well as dynamics on SO(3) typically fail to deliver results within common 1 ms iteration cycles. Therefore, they cannot be considered for the problem at hand. Representing approaches that adapt methods in $\mathbb{R}^6$ using Euler

Algorithm	**Reflexxes**	**Our approach**		
Representation	Euler ang. [Krö11]	Euler ang. [HGW17]	Rot. Matrix	Quat.
Iteration time	13.3 µs	8.3 µs	7.6 µs	5.1 µs

Table 2.2: Runtime Comparison for full 6D OTG

angle parametrization, we compare our algorithm with the available Reflexxes Motion Library [Ref13] that is based on the work of Kröger [Krö11]. For a fair comparison the freely available Type II version (unbound jerk) is used. The algorithm was compared in terms of runtime with the OTG of the available Reflexxes Motion Library. Both algorithms were analyzed using the Simulink 8.9 Profiler evaluating 10^6 iterations on a single 3.7 GHz CPU. The resulting computation are listed in Table 2.2. Although the implementation and compilation of our algorithm is not explicitly optimized for runtime, the computation time spent per iteration is below the Reflexxes Motion library for all three representations of 3D orientation in SO(3). This makes the proposed algorithm suitable as an OTG or an intermediate layer between an optional higher-level trajectory planner, that may run on a slower sample rate, and the robot controller. This increases robustness without large additional computational load.

2.6 Chapter Conclusion

In this chapter a new approach for an OTG was introduced. According to the classification by Kröger et al. [Krö10], our method describes an OTG of type II (i.e. bounded velocity and acceleration, unbounded jerk; allowing position and velocity targets) in variant B (i.e. time-variant constraints). It is directed – but not limited – to be used in HRC scenarios, where assuring human comfort and guaranteeing safety is prioritized over time optimality. While offering the typical advantages of OTG, such as increasing robustness against infeasible input trajectories (e.g. PTP step trajectories) and instantaneous reaction capabilities to unknown events, it offers intuitive definition of dynamic constraints. That is, constraining the magnitude of velocity and acceleration vectors of 3D translation as well as 3D rotation. This way the coordinate system independent dynamics of the end-effector, which can also be interpreted as kinetic energies on velocity level, are directly constrained. Further, rotations are constrained in their true geometric angular velocity, rather than purely analytical values such as direct derivation of Euler angles. Accurate time integration of the orientation differential equations is achieved by the means of the Magnus expansion.

This is the first real-time capable OTG algorithm to allow such constraints, while directly generating 3D translation and rotation trajectories on SE(3) in a singularity-free formulation. It was also outlined how the set of constraints can be extended to limit the velocity of the full robot structure in joint as well as 3D Euclidean space. Considering time-variance of these constraints opens up many possibilities to connect other metrics such as human-robot distance. Our work introduces an algorithm that provides the advantages of an indirect approach i.e. fast computation cycles and instantaneous reaction to unforeseen inputs. At the same time it also allows seamless transitions to directly forwarding of trajectories that already satisfy dynamic constraints. These transitions do not require an explicit switching or blending between different strategies, but directly result from the algorithm itself. measures to the OTG, as shown in experimental evaluation. The beauty of our algorithm lies clearly in its simplicity and its reduced and intuitive definition of the constraints, especially in terms of orientation. The verification of increasing acceptance by constraining magnitudes is still to be validated in user-case studies. Suggestions for future development are the extension jerk-limitation, which is of importance especially in industrial contexts. This requires a new step in the algorithm in which the correct acceleration profile has to be found before calculating the minimum time left.

"Stone, wood and iron are wrought and put together by mechanical methods, but the greatest work is to keep right the animal part of the machinery."

— John Smeaton (1724–1792)

3

Efficient Closed-Form Task Space Manipulability for a 7-DOF Serial Robot

This chapter was previously published in [HW20].

With the increasing demand for robots to react and adapt to unforeseen events, it is essential that a robot preserves agility at all times. While *manipulability* is a common measure to quantify agility at a given joint configuration, an efficient direct evaluation in task space is usually not possible with conventional methods, especially for redundant robots with an infinite number of inverse kinematics solutions. Yet, this is essential for global online optimization of a robot posture. In this chapter, we derive analytical expressions for a conventional 7-DOF serial robot structure, that enable direct evaluation of manipulability from a reduced task space parametrization. The resulting expressions allow array operation and thus achieve very high computational efficiency with vector-optimized programming languages. This direct and simultaneous calculation of the *task space manipulability* for large numbers of poses benefits many optimization problems in robotic applications. We show applications in global optimization of robot mounting poses, as well as redundancy resolution with global online optimization w.r.t. manipulability.

3.1 Introduction

It is a common requirement in robotic manipulation tasks to quantify the capabilities of a robot at a given pose. Having such a scalar measure allows comparison of different kinematic configurations in terms of the chosen metric, and can be considered at a path planning as well as at a control level. While these measures are usually defined in terms of a given joint configuration [Yos85b, Yos85a, CCSS92, Chi00, VAM$^+$12], the task of the robot is typically not given in this *joint space*. For a general robot the *task space* is usually defined in SE(3), i.e. the space of 3D poses consisting of translation and rotation. For many practical problems it is thus relevant to directly evaluate this measure w.r.t. a parametrization of SE(3) rather than the joints. This requires combining the evaluation of the inverse kinematics (IK) with the selected capability metric. But direct calculation of the IK is always robot dependent and general analytic solutions are not possible. This is especially true for redundant robots that have more DOFs in joint space than in task space and thus admit an infinite number of IK solutions for a given end-effector pose. While analytic IK solutions are well known for conventional 6-DOF kinematics [WS16], for general robotic structures numeric IK solvers are applied. However, they require several iterations to find an approximated joint configuration for a given end-effector pose. This is sufficient for calculating single poses, but it is inefficient for optimization problem solvers that require evaluation of large numbers of poses. This especially prevents time-critical computation of global optima. Expressions that can be evaluated directly, are thus superior for fast computation. While an analytical IK for a general robot structure does not exist, our work focuses on the most commonly used articulated 6- and 7-DOF robot serial kinematics. Yet, the 6 axis version can be viewed as a finite set of particular null-space solutions of the 7-DOF.

3.1.1 Contribution

In this chapter, we develop a set of computationally efficient closed-form expressions to evaluate the task space manipulability of a 7-DOF serial robot structure. The main contributions of this chapter consist of:

1. a new parametrization of the state- and null-space, that results in concise IK expressions with symmetric structure in the individual components
2. analytical closed-form expressions from task space to manipulability measure w.r.t. joint limits, that allow array operation in vector-optimized programming languages[1]
3. sensitivity analysis of manipulability in task space
4. real-time capable application for evaluating task space manipulability of the entire null-space, for globally optimal redundancy resolution w.r.t. manipulability of single poses and full trajectories on SE(3)

3.1.2 Related Work

For this concise review, we group previous work on the topic into the three areas: *1)* performance measures in robotics, *2)* direct methods for IK evaluation and algorithmic strategies on velocity level, and *3)* approaches for optimizing manipulability.

3.1.2.1 Performance Measures

The arguably most common performance measure for robot structures is the *manipulability* measure defined by Yoshikawa [Yos85b]. It is proportional to the volume of an ellipsoid, spanned by directional capabilities of a kinematic structure to generate velocities in task space at a given joint configuration. It is purely kinematic and does not consider any dynamic components. Yoshikawa also proposed a dynamic manipulability ellipsoid [Yos85a] on acceleration

[1] *Array operation* is also called *Vectorization* in e.g. MATLAB It refers to the exploitation of single instruction multiple data (SIMD) instructions of modern central processing units (CPUs) and allows to operate on multiple data points simultaneously.

level, for cases where dynamic effects cannot be neglected. This formulation was improved by Chiacchio et al. [CCSS92] to correctly account for gravity. A new formulation of a dynamic manipulability ellipsoid that better depicts the real manipulator capabilities in terms of task space accelerations was proposed by Chiacchio [Chi00].

Besides manipulability on velocity and acceleration level due to mere kinematic relations, it is essential for practical applications to also consider joint limits as constraints directly on position level. Vahrenkamp et al. [VAM+12] extended Yoshkawa's basic manipulability, by directly integrating joint limit penalization into the definition of the kinematic velocity Jacobian. This is achieved via a joint limit potential function.

Bong-Huan Jun et al. [JLK08] introduce a task-oriented manipulability measure. While Yoshikawa's original measure [Yos85b] denotes the manipulability of the whole manipulator system, [JLK08] considers manipulability w.r.t. sub-tasks that only affect parts of the task space, e.g. axis-specific tasks. Karim Abdel-Malek and Wei Yu [AYY04] proposed an alternative dexterity measure for robot placement, that does not depend on explicit IK solutions. They analyze an augmented Jacobian matrix that does not only hold information about position and orientation, but also joint limits of the end-effector. It represents the reachable workspace with surface patches and is computationally very demanding.

Our work has the aim of developing closed-form solutions that allow efficient array operation. For this reason, the task space manipulability formulation developed in this chapter applies Yoshikawa's original measure from [Yos85b]. Because its definition uses a determinant to map the joints to a scalar metric, and thus allows expansion to a continuous polynomial expression for efficient evaluation.

3.1.2.2 Inverse Kinematics

The IK problem of serial robot structures can be solved very elegantly on velocity level, due to the linear relation of joint and task space velocities. However, numeric integration of the resulting joint velocities to joint angles needs stabilization against numerical drift and thus results in an iterative scheme. Originally proposed by Wolovich and Elliot [WE84], this group of IK solvers is nowadays typically referred to as closed-loop inverse kinematics (CLIK) solvers. Colomé and Torras [CT14] give an overview of the most common CLIK solvers, with an additional experimental comparison in terms of convergence, numerical error, singularity handling, joint limit avoidance, and the capability of reaching secondary goals. Antonelli [Ant09] conducted a stability analysis of priority-based kinematic CLIK algorithms for redundant kinematics. He provides sufficient conditions for the control gains. While different stabilization schemes for CLIK solvers are proposed, the choice of gain parameters used in the control structure is rarely addressed. In practice these parameters are often empirically tuned. Bjoerlykhaug [Bjo18] proposes the use of a genetic algorithm for optimizing the feedback gain used in CLIK solvers, in order to minimize iteration cycles and maximize accuracy. In an experimental evaluation, he achieved a 50 % decrease in computation time through his feedback gain tuning. Reiter et al. [RMG18] propose a strategy for finding higher-order time-optimal IK solutions for redundant robots. They lay out solutions for fourth-order time derivatives of joint trajectories, applying a multiple shooting optimization method. This higher-order continuous differentiability is especially important for application on elastic mechanisms.

Siciliano [Sic90] gives a tutorial on early common online IK algorithms. He states the important features of a direct inverse kinematics function, i.e. repeatability, cyclicity or cyclic behavior, and online applicability. Shimizu et al. [SKY$^+$08] outline an analytical IK computation for a 7-DOF serial robot. The approach directly parametrizes the end-effector pose with Cartesian coordinates for translation and a rotation matrix for orientation. However, the use of the 2-quadrant atan function as opposed to the 4-quadrant atan2 function results in two problems. For one, not the entire task space is covered, and two, it results in discontinuous joint functions w.r.t. the null-space parameter and thus leads to discontinuous IK solutions and corresponding null-space limitations. A similar strategy, but extended to the entire domain, is proposed by Faria et al. [FFE$^+$18]. They propose a position-based IK solution for a 7-DOF serial manipulator with joint limit and singularity avoidance.

Besides approaches that use kinematic insight of a structure, also several machine learning algorithms are considered in literature. A detailed review is beyond the scope of this chapter, but we want to give a concise overview of research activities. D'Souza et al. [DVS01] apply a locally weighted projection regression to learn the IK of a 30-DOF humanoid robot. This maps the non-convex problem onto a locally convex problem that is suitable for direct learning. Tejomurtula and Kak [TK99], as well as Köker et al. [KÖÇE04], applied artificial neural networks for finding an IK mapping for 3-DOF robots and showed the feasibility of the problem using conventional error-backpropagation and Kohonen networks. Sariyildiz et al. [SUO$^+$12] compare support vector regression and artificial neural networks for learning IK mappings of a 7-DOF serial robot. They find that support vector regression is less prone to local minima and requires very few training data. Genetic algorithms were already early applied by Parker et al. [PKG89]. They pointed out low positioning accuracy, but emphasize its simplicity in application. Köker [Kök13] proposes a hybrid approach combining Elman neural networks with genetic algorithms. He was able to significantly improve accuracy for IK solutions of a 6-DOF mechanism in comparison to pure neural networks. Very recently, Dereli et al. [DK19] proposed a strategy to apply quantum behaved particle swarm optimization for finding IK solutions of a 7-DOF serial robot.

The IK expressions developed in this chapter are similar to the analytical approaches in [SKY$^+$08] and [FFE$^+$18] in terms of parametrizing the null-space as arm angle. However, the new task space parametrization that we introduce results in more concise and, more importantly, fully vectorizable expressions that allow efficient array operations. In contrast to existing approaches in literature, this computational advantage makes our approach suitable for simultaneous evaluation of a large number of poses.

3.1.2.3 Optimizing Manipulability

In conventional industrial contexts, optimizing cycling time is always of interest. Several publications deal with this problem, e.g. Kamrani et al. [KBW$^+$10] use the Response Surface Method [MMAC16] to optimize robot placement w.r.t. cycling time. Chan and Dubey [CD95], as well as Dariush et al. [DHO10], use a projection method of the joint limit gradient potential function. This is used for local manipulability optimization on velocity level. Dufour and Suleiman [DS17] present an approach of integrating the manipulability index into an optimization-based IK solver, by using linear approximations of the nonlinear manipulability measure with numeric gradient calculations at every time step. Jin et al. [JLLL17] mention the difficulty of real-time manipulability optimization that is related to a high computational burden since the manipulability is a non-convex function to the joint angles of a robotic arm. Due to the capability of *high-speed parallel distributed processing*, they propose an approach using dynamic neural networks in order to implement manipulability optimization in real-time. Conducting computer simulations, they show that the proposed method raises the manipulability by almost 40 % on average compared to existing methods.

Besides local optimization of a given joint configuration, for many robotic tasks it is required to include manipulability as criteria for optimization of the whole trajectory. Lee [Lee89] shows that a required motion can be approximated by a series of manipulability ellipsoids. Guilamo et al. [GKNK06] present an algorithm for trajectory generation that maximizes the volume of the manipulability ellipsoid. Yoshikawa [Yos85b] already observed that the optimal postures of various manipulators form the viewpoint of manipulability, often show resemblance of those naturally taken by human arms. This motivates the idea of manipulability transfer using a *learning by demonstration* strategy that is introduced by Rozo et al. [RJCC17]. Their approach allows robots to learn and reproduce a continuous set of manipulability ellipsoids by an expert's demonstration. In order to encode and retrieve those ellipsoids, they apply Gaussian Mixture Models and Gaussian Mixture Regression. In Jaquier et al. [JRCC18] the same authors exploit tensor-based representation, to consider that manipulability ellipsoids lie on the manifold of symmetric positive definite matrices. Faroni et al. [FBVT16] present an approach that maximizes the average manipulability of the overall task. Their method is based on the optimization of a cost function that depends on various points along a predetermined path. In particular, if the task of the manipulator is known *a priori*, this approach provides global manipulability optimization.

An approach for directly quantifying manipulability of a redundant robot in task space is proposed by Zacharias et al. [ZBWH13]. They introduce a capability map, to guide the decision on how to place a mobile robot relative to an object. It is a sampling-based approach, based on the manipulability index. While the approach reveals in which regions the robot is capable of grasping objects from different angles, the information of optimal approaching directions is lost.

The task space manipulability approach in this chapter enables for the first time global manipulability optimization with real-time capabilities, due to its efficient formulation.

3.1.3 Outline

The remainder of the chapter is organized as follows. The problem of a closed-loop task space manipulation framework is outlined in Section 3.2. In Section 3.3, the derivation of all analytical mappings is explained. Evaluation and analysis of the resulting task space manipulability is discussed in Section 3.4 and applied in global optimization formulations in Section 3.5. We conclude the chapter and outline future directions of development in Section 3.6.

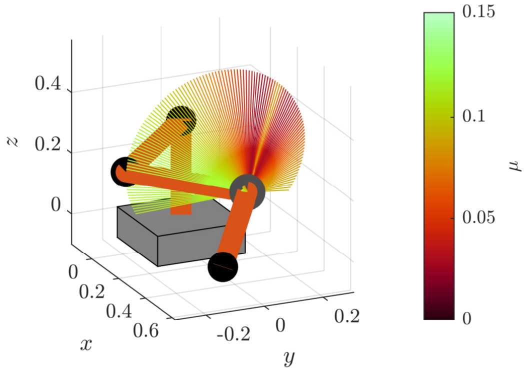


Figure 3.1: Illustration of the task space manipulability at a given end-effector pose. The null-space of this 7-DOF spherical-revolute-spherical (S-R-S) kinematic consists of the free elbow position (joint 4) along a circle. This position defines the direction of the forearm, i.e. the vector from the shoulder to the wrist. The coloured fan shows all possible forearm poses with the corresponding manipulability colour-coded from dark red (very bad) to light green (optimal). Colourless areas of the fan mark areas that violate joint constraints.

3.2 Problem Formulation

Given a n-DOF serial robot, its forward kinematics

$$\mathrm{FK}: \quad \mathbb{R}^n \to \mathrm{SE}(3) \times \mathbb{R}^{n-6}, \quad \boldsymbol{q} \mapsto (\boldsymbol{z}, \boldsymbol{\lambda}) \tag{3.1}$$

maps the joints $\boldsymbol{q}$ onto the 3D end-effector pose $\boldsymbol{z}$ at a particular null-space solution parametrized by $\boldsymbol{\lambda}$. To quantify the capability of moving in the SE(3) task space at a given joint configuration $\boldsymbol{q}$, a manipulability metric function

$$\mathrm{M}: \quad \mathbb{R}^n \to \mathbb{R}^1, \quad \boldsymbol{q} \mapsto \mu \tag{3.2}$$

is applied. A proper choice of parametrization for $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ assures the existence of the inverse function

$$\mathrm{IK}: \mathrm{SE}(3) \times \mathbb{R}^{n-6} \to \mathbb{R}^n, \quad (\boldsymbol{z}, \lambda) \mapsto \mathrm{FK}^{-1}(\boldsymbol{z}, \lambda) =: \boldsymbol{q}. \tag{3.3}$$

We define task space manipulability as the direct mapping

$$\mathrm{M} \circ \mathrm{IK}: \quad \mathrm{SE}(3) \times \mathbb{R}^{n-6} \to \mathbb{R}^1, \quad (\boldsymbol{z}, \boldsymbol{\lambda}) \mapsto \mu \tag{3.4}$$

of a desired pose $\boldsymbol{z}$ in task space onto the manipulability measure μ, considering all null-space solutions parametrized by $\boldsymbol{\lambda}$. $(\mathrm{M} \circ \mathrm{IK})(\boldsymbol{z}, \boldsymbol{\lambda})$ denotes the function composition $\mathrm{M}(\mathrm{IK}(\boldsymbol{z}, \boldsymbol{\lambda}))$. Figure 3.1 illustrates the task space manipulability for a certain end-effector pose $\boldsymbol{z}$. Considering real-time critical online applications and feasibility of global optimization formulations, the development of the task space manipulability mapping can be broken down into three problems:

Problem 1: Find a parametrization of the task- and null-space, that exploit the kinematic structure for concise expressions.

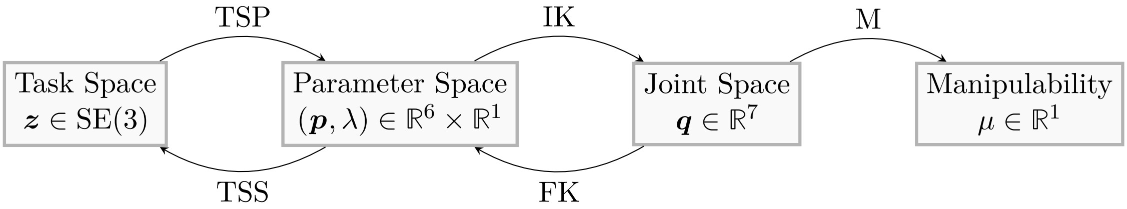

TSP
IK
M
Task Space
$\boldsymbol{z} \in \mathrm{SE}(3)$
Parameter Space
$(\boldsymbol{p}, \lambda) \in \mathbb{R}^6 \times \mathbb{R}^1$
Joint Space
$\boldsymbol{q} \in \mathbb{R}^7$
Manipulability
$\mu \in \mathbb{R}^1$
TSS
FK

Figure 3.2: Relation of task space $\boldsymbol{z}$, parameter space $\boldsymbol{p}$, joint space $\boldsymbol{q}$ and manipulability metric μ. The mappings are referred to as task space projection (TSP) and task space surjection (TSS), forward kinematics (FK) and inverse kinematics (IK), and manipulability (M).

Problem 2: Find closed-form expressions for all mappings from task space to manipulability that allow efficient array operation in vector-optimized programming languages.

Problem 3: Let $\mathcal{Q} \subset \mathbb{R}^7$ be the space of admissible joint configurations. Find an analytical expression of the range of the null-space solutions $\Lambda(\boldsymbol{z}) := \{\boldsymbol{\lambda} \in \mathbb{R}^{n-6} \mid \mathrm{IK}(\boldsymbol{z}, \boldsymbol{\lambda}) \in \mathcal{Q}\}$, for which the inverse kinematics function $\mathrm{IK}(\cdot, \boldsymbol{\lambda})$ results in an admissible joint configuration $\boldsymbol{q} \in \mathcal{Q}$.

In this chapter, we investigate in detail the case of a 7-DOF serial robot kinematics in conventional S-R-S structure,[2] such as the *KUKA LBR* series. This kinematic structure leads to a 1-dimensional null-space of solutions and thus $\lambda \in \mathbb{R}^1$.

3.3 Approach

This section outlines the derivation of the closed-form task space manipulability for the considered special case of a 7-DOF serial robot kinematics. We first discuss the chosen manipulability mapping and possible reductions in joint space. Motivated by these reductions, we propose a task space projection onto a parameter space, which yields concise expressions for the IK. Figure 3.2 summarizes all developed mappings that are developed in this section. The section concludes with an analytic definition of the admissible null-space at a given parameter end-effector pose.

3.3.1 Manipulability Measure

The central equation in robot kinematics is the linear forward velocity kinematic map

$$\dot{\boldsymbol{z}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) := \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.5}$$

that relates general joint space velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to task space velocities $\dot{\boldsymbol{z}} \in \mathbb{R}^6$. Where the linear map $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6 \times n}$ describes the velocity propagation from joint to task space at

[2] S-R-S refers to a kinematic 7-DOF structure with alternating revolute joints of which the rotation axes of the first and last 3 joints intersect. These two groups of intersecting axes behave kinematically like a spherical joint and are often referred to as shoulder and wrist.

a given joint configuration $\boldsymbol{q} \in \mathbb{R}^n$. It is defined by the kinematic chain and represents the derivative

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \dot{\boldsymbol{z}}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}, \tag{3.6}$$

hence it is often referred to as *Robot Jacobian*.

Yoshikawa's manipulability measure [Yos85b], that we use in this work, is defined as

$$\mathrm{M}: \quad \mathbb{R}^n \to \mathbb{R}, \quad \boldsymbol{q} \mapsto \sqrt{\det\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^\top\right)} =: \mu \tag{3.7}$$

and is a measure, proportional to the volume of the velocity manipulability ellipsoid

$$\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{q}} = 1 \tag{3.8a}$$

$$\dot{\boldsymbol{z}}^\top (\boldsymbol{J}\boldsymbol{J}^\top)^{-1} \dot{\boldsymbol{z}} = 1. \tag{3.8b}$$

Note that (3.7) does not consider hardware-related joint limits. However, joint configurations that violate these constraints must not be considered.

Zlatanov et al. [ZFB94] explain that the forward velocity kinematic map (3.5) is not sufficient for exhaustive characterization of the singularities of a manipulator. Further, Staffetti et al. [SBDS02] show that many of these often-used manipulability indices are not invariant to change of reference frames, scale or physical units. However, the big advantage of Yoshikawa's original manipulability metric is the fact that it can be expanded to a polynomial expression and thus qualifies for computationally efficient array operation. Further, derivatives can be calculated analytically. As outlined by Staffetti et al. [SBDS02], it is not a true metric for distance to a singularity but nonetheless serves as a relative comparison of manipulability qualities between joint configurations [GT04].

For a n-DOF serial robot kinematics, we refer to the $i = [1, n]$ absolute angular and translational velocities of the individual links, i.e. the velocity between the robot base B and the body-fixed frame of link i, as $\boldsymbol{\omega}_{Bi}$ and $\boldsymbol{v}_{Bi}$. Expressed w.r.t. the link frame i, the velocities of the kinematic chain are calculated with

$${}_i\boldsymbol{\omega}_{Bi} = \boldsymbol{R}_{ip}\, {}_p\boldsymbol{\omega}_{Bp} + {}_i\boldsymbol{\omega}_{pi} \tag{3.9a}$$

$${}_i\boldsymbol{v}_{Bi} = \boldsymbol{R}_{ip}\left({}_p\boldsymbol{v}_{Bp} + {}_p\boldsymbol{\omega}_{Bp} \times {}_p\boldsymbol{r}_{pi}\right), \tag{3.9b}$$

where $p = i - 1$ is the predecessor link of i and $(\times)$ denotes the cross product $\mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$. In the following, *manipulability* refers to Yoshikawa's manipulability measure [Yos85b].

3.3.1.1 Reduction of First Joint

While Yoshikawa's manipulability measure is not invariant w.r.t. scale or physical units, it is in fact invariant to change of reference frames.

Proof. Given a vector of joint velocities $\dot{\boldsymbol{q}}$ and task space velocities $\dot{\boldsymbol{z}}$ w.r.t. a reference frame X, the Jacobian matrix

$$\boldsymbol{J}_X(\boldsymbol{q}) = \frac{\partial\, {}_X\dot{\boldsymbol{z}}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}} \tag{3.10}$$

is used to define the manipulability index

$${}_X\mu(\boldsymbol{q}) = \sqrt{\det\left(\boldsymbol{J}_X(\boldsymbol{q})\boldsymbol{J}_X(\boldsymbol{q})^\top\right)}. \tag{3.11}$$

If this manipulability index is expressed in terms of a new reference frame Y via the block transformation matrix

$$\boldsymbol{R}_{YX}^{\text{block}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R}_{YX}(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{YX}(\boldsymbol{q}) \end{bmatrix}, \tag{3.12}$$

consisting of rotation matrices $\boldsymbol{R}_{YX}$, the manipulability index reads

$${}_Y\boldsymbol{\mu}(\boldsymbol{q}) = \sqrt{\det\left(\boldsymbol{R}_{YX}^{\text{block}}(\boldsymbol{q})\boldsymbol{J}_X(\boldsymbol{q})\left(\boldsymbol{R}_{YX}^{\text{block}}(\boldsymbol{q})\boldsymbol{J}_X(\boldsymbol{q})\right)^\top\right)}. \tag{3.13}$$

Considering the fact that Euclidean transformation matrices have $\det(\boldsymbol{R}) = 1$, we find

$${}_Y\boldsymbol{\mu}(\boldsymbol{q}) = \sqrt{\det\left(\boldsymbol{J}_X(\boldsymbol{q})\boldsymbol{J}_X(\boldsymbol{q})^\top\right)} = {}_X\boldsymbol{\mu}(\boldsymbol{q}) \tag{3.14}$$

i.e. the manipulability measure μ is invariant to change of reference frames. □

If the reference frame is chosen to be fixed to any link after the first joint, it results in an expression for the manipulability measure that is independent of the first joint. This results from the fact that the first joint rotates the whole kinematic structure including the reference frame, but does not alter any geometric relations.

We consequently choose to formulate the Jacobian matrix w.r.t. the end-effector frame, as this does not only lead to the independence of q_1, but also results in the most concise expression.

3.3.1.2 Reduction of Last Joint

For a special case of a 7-DOF serial kinematic, the parameter space of the manipulability can be further reduced. This special case consists of kinematic structures, whose origin of the end-effector frame lies on the rotation axis of the last joint q_n. The purely angular contribution of q_n does not alter the kinematic configuration but only rotates the reference frame and with it the manipulability ellipsoid. The shape of the ellipsoid is not affected and so q_n can also not influence the manipulability measure.

3.3.1.3 Closed-Form Expression

Exploiting these two reductions by formulating the $\boldsymbol{J}_T$ w.r.t. the end-effector frame T and assuming the TCP along the last joint axis, it is possible to expand the entire determinant expression of the matrix $\boldsymbol{J}_T \boldsymbol{J}_T{}^\top \in \mathbb{R}^{6\times 6}$ from (3.7) to a symbolic polynomial expression using e.g. *MATLAB Symbolic Math Toolbox™*. The advantage being, that unlike the original matrix expression, the polynomial form allows array operation in vector-optimized programming languages. This enables simultaneous evaluation of an entire set of joint configurations. The full manipulability function is listed in Appendix B.1.

3.3.2 Task Space Parametrization

The decision of choosing a parametrization for the SE(3) pose as well as the 1D null-space, is essential for the derivation of concise analytical formulations. We propose the following parameter requirements (PR) for a suitable parametrization in regard to the IK functions. The parameter set must

PR1: uniquely define the null-space parameter for the entire space of SE(3).

PR2: result in a minimal number of parameters for the components of the IK vector map $\boldsymbol{p} \mapsto \boldsymbol{q}$.

PR3: allow direct application of the above-mentioned reductions.

Different approaches for null-space parametrization were proposed in literature. The redundancy is either directly parametrized by a redundant joint [LB91, Ton06], or more commonly by a joint independent arm angle [KDLS92, SKY+08]. Shimizu et al. [SKY+08] argued, that joint-based parametrization is not suitable for the discussed 7-DOF S-R-S mechanism due to possible ambiguous results. Kreutz-Delgado et al. [KDLS92] define the arm angle as the angle between an arm and a reference plane. The arm plane is spanned by shoulder, elbow and wrist locations. The reference plane is defined by a fixed vector and the vector from shoulder to wrist. Shimizu et al. [SKY+08] point out arithmetic singularities in the original definition whenever the two vectors are collinear. They enhance the robustness of the definition by defining the reference plane in terms of a particular solution $q_3 \overset{!}{=} 0$, that resembles the solution of conventional non-redundant 6-DOF mechanisms. While this definition is unique w.r.t. the conservative joint limits of their analyzed robot structure, it is ambiguous whenever the reduced non-redundant 6-DOF mechanism admits multiple configurations that result in the same end-effector pose.

In this chapter, we introduce a parametrization that fulfills all the above-discussed parameter requirements. Figure 3.3 illustrates the following discussion. Independent of a desired end-effector pose, positions of the base B and shoulder S are always stationary, where

$$ {}_B\boldsymbol{r}_{BS} := (l_B + l_1)\hat{\boldsymbol{z}} \tag{3.15} $$

with link lengths of the base link l_B and the first link l_1. Additionally defining a desired end-effector pose relative to the robot base in SE(3), consisting of ${}_B\boldsymbol{r}_{BT}$ for translation and

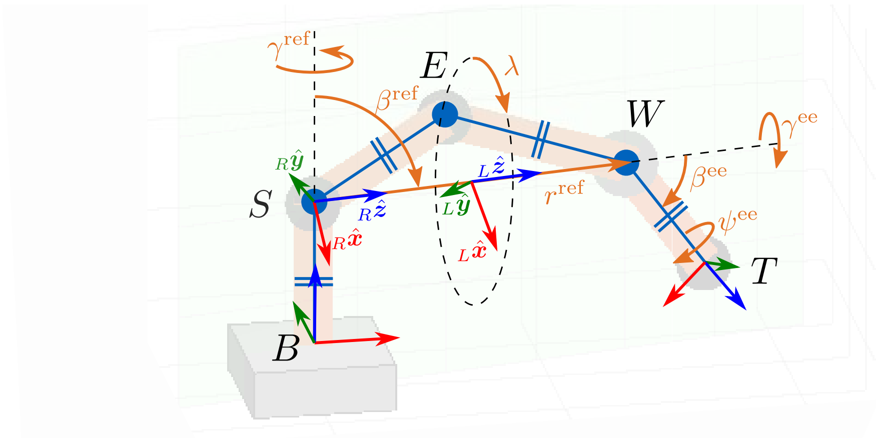

γ^{ref}
E
λ
β^{ref}
W
γ^{ee}
${}_R\hat{\boldsymbol{y}}$
${}_L\hat{\boldsymbol{z}}$
r^{ref}
β^{ee}
S
${}_R\hat{\boldsymbol{z}}$
${}_L\hat{\boldsymbol{y}}$
ψ^{ee}
${}_R\hat{\boldsymbol{x}}$
${}_L\hat{\boldsymbol{x}}$
T
B

Figure 3.3: Parametrization of the Task Space. Positions of Base B and Shoulder S are fixed. Translation reference parameters $(r^{\mathrm{ref}}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}})$ define the position of the Wrist W. The end-effector parameters $(\gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \psi^{\mathrm{ee}})$ describe the rotation from reference frame R to tool frame T as consecutive $Z \rightarrow Y^{'} \rightarrow Z^{''}$ Euler angles. The null-space is parametrized with λ. It defines the position of the elbow E via relative rotation between the elbow oriented frame L and frame R.

$\boldsymbol{R}_{TB}$ for orientation, determines not only the location of the tool-center-point T but also the wrist position

$$ {}_B\boldsymbol{r}_{BW} := {}_B\boldsymbol{r}_{BT} - \boldsymbol{R}_{B6}(l_6 + l_7 + l_T)\hat{\boldsymbol{z}}, \tag{3.16}$$

with link lengths l_6 and l_7, and a potential tool length l_T. This wrist position is used for define the translational component of the end-effector pose $\boldsymbol{z}$. The position ${}_B\boldsymbol{r}_{SW}$ is parametrized by spherical coordinates $(r^{\mathrm{ref}}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}})$ with coordinate plane ${}_B\hat{\boldsymbol{x}}\hat{\boldsymbol{z}}$, origin S and ${}_B\hat{\boldsymbol{z}}$ as polar axis. The parameters are radius r^{ref}, longitudinal angle γ^{ref}, and azimuthal angle β^{ref}. Note that γ^{ref} and β^{ref} directly align with the rotation axis of q_1 and q_2. These two angles also define the reference frame R with

$$\boldsymbol{R}_{RB}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}) := \boldsymbol{R}_y(\beta^{\mathrm{ref}})\boldsymbol{R}_z(\gamma^{\mathrm{ref}}). \tag{3.17}$$

The orientation is parametrized along a consecutive Euler angle sequence $Z \rightarrow Y^{'} \rightarrow Z^{''}$, which again corresponds to the sequence of the joint structure. However, instead of directly parametrizing $\boldsymbol{R}_{TB}$, we parametrize the end-effector orientation with respect to the reference frame, i.e.

$$\boldsymbol{R}_{TR}(\gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \psi^{\mathrm{ee}}) := \boldsymbol{R}_z(\psi^{\mathrm{ee}})\boldsymbol{R}_y(\beta^{\mathrm{ee}})\boldsymbol{R}_z(\gamma^{\mathrm{ee}}). \tag{3.18}$$

Regarding the stated parameter requirement *PR2*, this makes the IK functions of the wrist angles (q_5, q_6, q_7) as independent of the shoulder parameters $(r^{\mathrm{ref}}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}})$ as possible, as will be seen in the IK Section 3.3.3.

The 1D null-space is parametrized by the arm angle λ. In contrast to Shimizu et al. [SKY$^+$08] we do not define the arm angle w.r.t. the non-redundant solution $q_3 \overset{!}{=} 0$, but w.r.t. to the introduced reference frame R. Let λ be the arm angle, that defines a new frame L with

$$\boldsymbol{R}_{LB}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}, \lambda) := \boldsymbol{R}_z(\lambda)\boldsymbol{R}_{RB}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}), \tag{3.19}$$

such that the negative frame base vector $(-{}_L\hat{\boldsymbol{x}})$ points in direction of the elbow E. This uniquely defines the null-space parameter as required in *PR1*. The full set of parameters is thus given with tuple $(\boldsymbol{p}, \lambda) \in \mathbb{R}^6 \times \mathbb{R}$, consisting of the parameter vector

$$\boldsymbol{p} := \left[r^{\text{ref}}, \gamma^{\text{ref}}, \beta^{\text{ref}}, \gamma^{\text{ee}}, \beta^{\text{ee}}, \psi^{\text{ee}}\right]^\top \tag{3.20}$$

and arm angle λ. The individual parameter range definitions are

$$\begin{aligned}
r^{\text{ref}} &\in [\;\; r^{\text{ref}}_{\text{min}}, \;\; r^{\text{ref}}_{\text{max}} \;\;] \\
\gamma^{\text{ref}} &\in [\;\; -\pi, \;\; +\pi \;\;] \\
\beta^{\text{ref}} &\in [\;\; 0, \;\; +\pi \;\;] \\
\gamma^{\text{ee}} &\in [\;\; -\pi, \;\; +\pi \;\;] \\
\beta^{\text{ee}} &\in [\;\; 0, \;\; +\pi \;\;] \\
\psi^{\text{ee}} &\in [\;\; -\pi, \;\; +\pi \;\;] \\
\lambda &\in [\;\; -\pi, \;\; +\pi \;\;]
\end{aligned} \tag{3.21}$$

and form the parameter space $\mathcal{P} \subset \mathbb{R}^7$. Note that the two parameters γ^{ref} and ψ^{ee} solely affect joints q_1 and q_7, which do not influence manipulability. The task space manipulability developed in this chapter can thus without loss of information be represented by the reduced parameter vector $\boldsymbol{p}^{\text{red}} \in \mathcal{P}^{\text{red}} \subset \mathbb{R}^4$ consisting of

$$\boldsymbol{p}^{\text{red}} := \left[r^{\text{ref}}, \beta^{\text{ref}}, \gamma^{\text{ee}}, \beta^{\text{ee}}\right]^\top . \tag{3.22}$$

This complies with the stated requirement *PR3*. The presented parametrization is the fundamental core for the concise mappings developed in the remaining section.

3.3.2.1 Task Space Projection

We refer to the extraction of the parameter vector $\boldsymbol{p} = \left[r^{\text{ref}}, \gamma^{\text{ref}}, \beta^{\text{ref}}, \gamma^{\text{ee}}, \beta^{\text{ee}}, \psi^{\text{ee}}\right]^\top$ from a given end-effector pose $\boldsymbol{z} \in \text{SE}(3)$ as *task space projection*. Without loss of generality, we assume the pose $\boldsymbol{z} \in \text{SE}(3)$ is described with Cartesian Coordinates (x, y, z) for translation ${}_B\boldsymbol{r}_{BT}$ together with a Rotation matrix $\boldsymbol{R}_{TB}$ for orientation. As a reference matrix for extracting the parameter space angles $(\gamma^{\text{ee}}, \beta^{\text{ee}}, \psi^{\text{ee}})$ we state the rotation matrix for a general *ZYZ* Euler sequence[3]

$$\boldsymbol{R}_{zyz}(\gamma, \beta, \psi) := \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\gamma) =$$
$$\begin{pmatrix} \mathrm{c}(\beta)\,\mathrm{c}(\gamma)\,\mathrm{c}(\psi) - \mathrm{s}(\gamma)\,\mathrm{s}(\psi) & \mathrm{c}(\gamma)\,\mathrm{s}(\psi) + \mathrm{c}(\beta)\,\mathrm{c}(\psi)\,\mathrm{s}(\gamma) & -\mathrm{c}(\psi)\,\mathrm{s}(\beta) \\ -\mathrm{c}(\psi)\,\mathrm{s}(\gamma) - \mathrm{c}(\beta)\,\mathrm{c}(\gamma)\,\mathrm{s}(\psi) & \mathrm{c}(\gamma)\,\mathrm{c}(\psi) - \mathrm{c}(\beta)\,\mathrm{s}(\gamma)\,\mathrm{s}(\psi) & \mathrm{s}(\beta)\,\mathrm{s}(\psi) \\ \mathrm{c}(\gamma)\,\mathrm{s}(\beta) & \mathrm{s}(\beta)\,\mathrm{s}(\gamma) & \mathrm{c}(\beta) \end{pmatrix} \tag{3.23}$$

that shows that we can define a mapping eul_{ZYZ} as

$$\text{eul}_{ZYZ}: \quad \text{SE}(3) \to \mathbb{R}^3, \quad \boldsymbol{R}_{zyz} \mapsto \begin{bmatrix} \text{atan2}\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,2)}\,,\, [\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,1)}\right) \\ \arccos\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,3)}\right) \\ \text{atan2}\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(2,3)}\,,\, -[\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(1,3)}\right) \end{bmatrix} =: \begin{bmatrix} \gamma \\ \beta \\ \psi \end{bmatrix} \tag{3.24}$$

[3] we use s(·) and c(·) to shorten the notation of the sin (·) and cos (·) functions resp.

that extracts the Euler angles from a rotation matrix in SE(3). The operator $[\,\cdot\,]_{(i,j)}$ returns the element at row i and column j of a matrix.

The task space projection

$$\mathrm{TSP}: \quad \mathrm{SE}(3) \to \mathbb{R}^6, \quad \boldsymbol{z} \mapsto \boldsymbol{p} \tag{3.25a}$$

consists of the mappings

$$r^{\mathrm{ref}}(\boldsymbol{z}) := \|\,{}_B\boldsymbol{r}_{SW}\|_2 \tag{3.25b}$$

$$\beta^{\mathrm{ref}}(\boldsymbol{z}) := \frac{\pi}{2} - \arctan\frac{[\,{}_B\boldsymbol{r}_{SW}]_{(3)}}{[\,{}_B\boldsymbol{r}_{SW}]_{(1)}} \tag{3.25c}$$

$$\gamma^{\mathrm{ref}}(\boldsymbol{z}) := \operatorname{atan2}\left(\Big[\,{}_B\boldsymbol{r}_{SW}\Big]_{(2)}, \Big[\,{}_B\boldsymbol{r}_{SW}\Big]_{(1)}\right) \tag{3.25d}$$

$$\begin{bmatrix}\gamma^{\mathrm{ee}}\\ \beta^{\mathrm{ee}}\\ \psi^{\mathrm{ee}}\end{bmatrix}(\boldsymbol{z}) := \mathrm{eul}_{ZYZ}(\boldsymbol{R}_{7R}(\boldsymbol{z}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}})). \tag{3.25e}$$

With the shoulder-wrist vector

$$\begin{aligned} {}_B\boldsymbol{r}_{SW} &:= {}_B\boldsymbol{r}_{BR} - {}_B\boldsymbol{r}_{BS} \\ &= {}_B\boldsymbol{r}_{BT} - \boldsymbol{R}_{B6}\,{}_6\hat{\boldsymbol{z}}(l_6 + l_7 + l_T) - {}_B\hat{\boldsymbol{z}}(l_B + l_1). \end{aligned} \tag{3.26}$$

and the rotation matrix

$$\boldsymbol{R}_{7R}(\boldsymbol{z}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}) := \boldsymbol{R}_{7T}\boldsymbol{R}_{TB}(\boldsymbol{z})\boldsymbol{R}_{BR}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}), \tag{3.27}$$

derived from the desired task space pose. Rotation $\boldsymbol{R}_{7T}$ is the constant rotation matrix from body fixed frame of link 7 to the TCP frame.

3.3.2.2 Task Space Surjection

We refer to the inverse mapping, i.e. from the parameter vector $\boldsymbol{p}$ to the task space pose $\boldsymbol{z}$, as task space surjection (TSS)

$$\mathrm{TSS}: \quad \mathbb{R}^6 \to \mathrm{SE}(3), \quad \boldsymbol{p} \mapsto \boldsymbol{z}. \tag{3.28a}$$

The relations are given with

$${}_B\boldsymbol{r}_{BT} := (l_B + l_1)\,{}_B\hat{\boldsymbol{z}} + \boldsymbol{R}_{BR}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}})\left({}_R\hat{\boldsymbol{z}}r^{\mathrm{ref}} + \boldsymbol{R}_{R7}(\gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \psi^{\mathrm{ee}})(l_6 + l_7 + \boldsymbol{R}_{7T}l_T)\right) \tag{3.28b}$$

$$\boldsymbol{R}_{TB} := \boldsymbol{R}_{7T}\boldsymbol{R}_{7R}(\gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \psi^{\mathrm{ee}})\boldsymbol{R}_{RB}(\gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}) \tag{3.28c}$$

using the established definitions in the previous sections.

3.3.3 Inverse Kinematics

In this section we derive closed-form expressions for the IK map. After discussing the choice of the default manipulator configuration, we derive the individual IK mappings of the robot joints. Corresponding to the S-R-S structure, we group the joints into shoulder angles $\{q_1, q_2, q_3\}$, the elbow angle $\{q_4\}$, and wrist angles $\{q_5, q_6, q_7\}$.

3.3.3.1 Manipulator Configuration

Due to the possible reconfiguration of the robot kinematics, i.e. whenever 3 revolute joint axes intersect in one point, with 2 being coaxial and the third being perpendicular to the links, there exists an alternative configuration

$$\mathrm{FK}(\mathrm{coaxial}_1, \mathrm{perpendicular}, \mathrm{coaxial}_2) = \mathrm{FK}(\mathrm{coaxial}_1 + \pi, -\mathrm{perpendicular}_2, \mathrm{coaxial}_2 + \pi) \tag{3.29}$$

that results in the same FK. In the 7-DOF S-R-S structure considered in this work, this is the case for the tuples (q_1, q_2, q_3), (q_3, q_4, q_5) and (q_5, q_6, q_7). Therefore, defining only the end-effector pose as well as the elbow position results in 8 possible configurations. Of course it is important to derive an IK map, that results in one specific configuration for the entire parameter space. The following derivation is designed to yield in a configuration as depicted in Figure 3.3 for the default case $q_1 = q_3 = q_5 = 0$. This is achieved by choosing the joint angle ranges

$$\begin{aligned} q_1 &\in [\ -\pi, \ +\pi\ ] \\ q_2 &\in [\ 0, \ +\pi\ ] \\ q_3 &\in [\ -\pi, \ +\pi\ ] \\ q_4 &\in [\ 0, \ +\pi\ ] \\ q_5 &\in [\ -\pi, \ +\pi\ ] \\ q_6 &\in [\ 0, \ +\pi\ ] \\ q_7 &\in [\ -\pi, \ +\pi\ ]. \end{aligned} \tag{3.30}$$

We refer to this definition as $\mathcal{Q}^{\mathrm{sc}} \subset \mathbb{R}^7$, i.e. the space of joints in standard configuration.

3.3.3.2 Elbow Angles

The central geometric shape to express the arm portion of joints is the triangle $\overline{SEW}$ as depicted in Figure 3.3. It is fully defined by the parameter r^{ref}, as well as the robot related constant link lengths

$$r_{SE} := l_3 + l_4 \tag{3.31}$$

$$r_{EW} := l_5 + l_6. \tag{3.32}$$

The law of cosines in this triangle allows direct calculation of joint 4

$$\left(r^{\mathrm{ref}}\right)^2 = (r_{SE})^2 + (r_{EW})^2 - r_{SE} r_{EW} \cos(\pi - q_4) \tag{3.33}$$

$$q_4(r^{\mathrm{ref}}) := \pi - \arccos\left(\frac{(r_{SE})^2 + (r_{EW})^2 - \left(r^{\mathrm{ref}}\right)^2}{2 r_{SE} r_{EW}}\right) \tag{3.34}$$

as well as the adjoint angles

$$r_{SE} = \left(r^{\mathrm{ref}}\right)^2 + (r_{EW})^2 - r^{\mathrm{ref}} r_{EW} \cos(\theta_S) \tag{3.35}$$

$$\theta_S(r^{\mathrm{ref}}) := \arccos\left(\frac{\left(r^{\mathrm{ref}}\right)^2 + (r_{EW})^2 - r_{SE}}{2 r^{\mathrm{ref}} r_{EW}}\right) \tag{3.36}$$

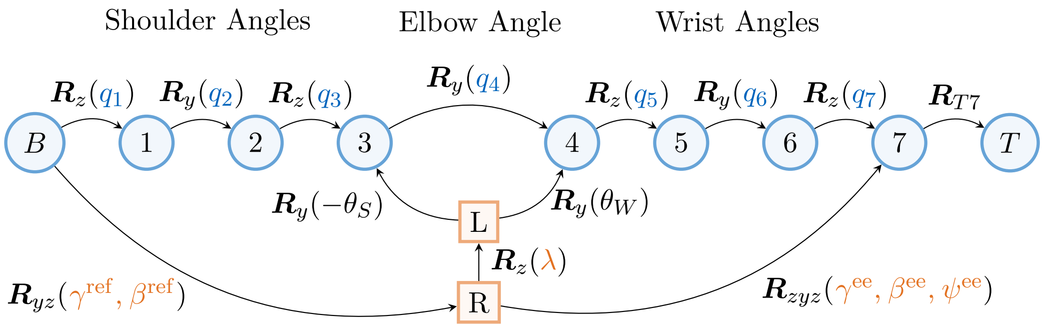


Figure 3.4: Reference frames and their relations. The blue frames B to T are fixed to the corresponding body-fixed coordinate systems of the robot links. Orange frames R and L are additional reference frames for the introduced parameter space. The arrows mark the rotations between the frames of reference.

and

$$r_{EW} = \left(r^{\text{ref}}\right)^2 + (r_{SE})^2 - r^{\text{ref}} r_{SE} \cos(\theta_W) \tag{3.37}$$

$$\theta_W(r^{\text{ref}}) := \arccos\left(\frac{\left(r^{\text{ref}}\right)^2 + (r_{SE})^2 - r_{EW}}{2 r^{\text{ref}} r_{SE}}\right). \tag{3.38}$$

The latter are used to define alternative rotation frame compositions for the derivation of the remaining joints. See Figure 3.4 for an overview of the relations between all introduced coordinate frames.

3.3.3.3 Shoulder Angles

Reusing the *ZYZ* Euler sequence extraction function (3.24), makes it possible to directly define the IK function of the shoulder angles $\{q_1, q_2, q_3\}$. The parameter related frames R and L (cf. Figure 3.4) are used to compose the transformation matrix

$$\boldsymbol{R}_{3B}(\boldsymbol{p}, \lambda) := \boldsymbol{R}_y(-\theta_W)\, \boldsymbol{R}_z(\lambda)\, \boldsymbol{R}_{RB}\left(\beta^{\text{ref}}, \gamma^{\text{ref}}\right) \tag{3.39}$$

and extract

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} (\boldsymbol{p}, \lambda) := \text{eul}_{ZYZ}\left(\boldsymbol{R}_{3B}(\boldsymbol{p}, \lambda)\right). \tag{3.40}$$

3.3.3.4 Wrist Angles

Analog to the shoulder angles, the wrist angles $\{q_5, q_6, q_7\}$ can be calculated by composing the transformation matrix

$$\boldsymbol{R}_{74}(\boldsymbol{p}, \lambda) := \boldsymbol{R}_{7R}\left(\gamma^{\text{ee}}, \beta^{\text{ee}}, \psi^{\text{ee}}\right) \boldsymbol{R}_z(-\lambda) \boldsymbol{R}_y\left(-\theta_S\right) \tag{3.41}$$

and extracting the wrist angles with

$$\begin{bmatrix} q_5 \\ q_6 \\ q_7 \end{bmatrix} (\boldsymbol{p}, \lambda) := \mathrm{eul}_{ZYZ} \left(\boldsymbol{R}_{74}(\boldsymbol{p}) \right). \tag{3.42}$$

3.3.3.5 Overview

All closed-form expressions resulting from the IK mapping are fully listed in Section B.2. The parameter dependencies of the individual function components are

$$\begin{aligned}
\mathrm{IK}_1 : \quad & \mathbb{R}^4 \to \mathbb{R}^1, & (\theta_\mathrm{S}(r^\mathrm{ref}), \gamma^\mathrm{ref}, \beta^\mathrm{ref}, \lambda) &\mapsto q_1 && \text{(3.43a)}\\
\mathrm{IK}_2 : \quad & \mathbb{R}^3 \to \mathbb{R}^1, & (\theta_\mathrm{S}(r^\mathrm{ref}), \beta^\mathrm{ref}, \lambda) &\mapsto q_2 && \text{(3.43b)}\\
\mathrm{IK}_3 : \quad & \mathbb{R}^3 \to \mathbb{R}^1, & (\theta_\mathrm{S}(r^\mathrm{ref}), \beta^\mathrm{ref}, \lambda) &\mapsto q_3 && \text{(3.43c)}\\
\mathrm{IK}_4 : \quad & \mathbb{R}^1 \to \mathbb{R}^1, & (r^\mathrm{ref}) &\mapsto q_4 && \text{(3.43d)}\\
\mathrm{IK}_5 : \quad & \mathbb{R}^3 \to \mathbb{R}^1, & (\theta_\mathrm{W}(r^\mathrm{ref}), \gamma^\mathrm{ee}, \beta^\mathrm{ee} - \lambda) &\mapsto q_5 && \text{(3.43e)}\\
\mathrm{IK}_6 : \quad & \mathbb{R}^3 \to \mathbb{R}^1, & (\theta_\mathrm{W}(r^\mathrm{ref}), \gamma^\mathrm{ee}, \beta^\mathrm{ee} - \lambda) &\mapsto q_6 && \text{(3.43f)}\\
\mathrm{IK}_7 : \quad & \mathbb{R}^4 \to \mathbb{R}^1, & (\theta_\mathrm{W}(r^\mathrm{ref}), \gamma^\mathrm{ee}, \beta^\mathrm{ee}, \psi^\mathrm{ee} - \lambda) &\mapsto q_7 && \text{(3.43g)}
\end{aligned}$$

and show the low dimensional dependency as required by *PR2*. Note that parameters γ^ref and ψ^ee do solely influence q_1 and q_7 resp., and thus do not influence manipulability. Further, in this formulation the shoulder and wrist joints result in equivalent mappings, with symmetrical assignments. Their relations are given as

$$\begin{aligned}
\mathrm{IK}_5 &= \mathrm{IK}_3(\theta_\mathrm{W}(r^\mathrm{ref}), \beta^\mathrm{ee}, \gamma^\mathrm{ee} - \lambda) && \text{(3.44a)}\\
\mathrm{IK}_6 &= \mathrm{IK}_2(\theta_\mathrm{W}(r^\mathrm{ref}), \beta^\mathrm{ee}, \gamma^\mathrm{ee} - \lambda) && \text{(3.44b)}\\
\mathrm{IK}_7 &= \mathrm{IK}_1(\theta_\mathrm{W}(r^\mathrm{ref}), \psi^\mathrm{ee}, \beta^\mathrm{ee}, \gamma^\mathrm{ee} - \lambda). && \text{(3.44c)}
\end{aligned}$$

This is an interesting geometrical insight that results from the chosen parameter set.

3.3.4 Forward Kinematics

Although not used in the task space manipulability mapping, we state – for the sake of completeness – the forward mapping

$$\mathrm{FK}: \quad \mathbb{R}^7 \to \mathbb{R}^6 \times \mathbb{R}, \quad \boldsymbol{q} \mapsto (\boldsymbol{p}, \lambda) \tag{3.45}$$

using the developed relations from the previous section on the IK problem. From the elbow angle q_4 and the relation (3.33), r^ref is mapped by

$$r^\mathrm{ref}(q_4) := \sqrt{(r_{SE})^2 + (r_{EW})^2 - r_{SE} r_{EW} \cos(\pi - q_4)}. \tag{3.46}$$

The Euler angle extraction function (3.24) allows again a concise definition of the remaining mappings. The shoulder joints $\{q_1, q_2, q_3\}$ with the adjoint shoulder angle $\theta_\mathrm{S}(r^\mathrm{ref})$ from (3.35) parametrize

$$\begin{bmatrix} \gamma^\mathrm{ref} \\ \beta^\mathrm{ref} \\ \lambda \end{bmatrix} := \mathrm{eul}_{ZYZ} \left(\boldsymbol{R}_{LB}(\boldsymbol{q}, r^\mathrm{ref}) \right) \tag{3.47a}$$

where

$$\boldsymbol{R}_{LB}(\boldsymbol{q}, r^{\text{ref}}) := \boldsymbol{R}_y(\theta_{\text{S}}(r^{\text{ref}}))\boldsymbol{R}_z(q_3)\boldsymbol{R}_y(q_2)\boldsymbol{R}_z(q_1). \tag{3.47b}$$

Analogously, the wrist joints $\{q_5, q_6, q_7\}$ and the adjoint wrist angle $\theta_{\text{W}}(r^{\text{ref}})$ from (3.37) define the end-effector parameters

$$\begin{bmatrix} \lambda + \gamma^{\text{ee}} \\ \beta^{\text{ee}} \\ \psi^{\text{ee}} \end{bmatrix} := \text{eul}_{ZYZ}\left(\boldsymbol{R}_{7L}(\boldsymbol{q}, r^{\text{ref}})\right) \tag{3.48a}$$

where

$$\boldsymbol{R}_{7L}(\boldsymbol{q}, r^{\text{ref}}) := \boldsymbol{R}_z(q_7)\boldsymbol{R}_y(q_6)\boldsymbol{R}_z(q_5)\boldsymbol{R}_y(-\theta_{\text{W}}(r^{\text{ref}})). \tag{3.48b}$$

The composition of rotations is in accordance with the structural relation depicted in Figure 3.4. This concludes the FK problem.

3.3.5 Admissible Parameter Space

The compact analytical expressions also allow solving analytically for an upper and lower bound of λ, given maximal joint angles $q_i^{\max}$. Let $\mathcal{Q} := \{\boldsymbol{q} \mid \boldsymbol{q} \in \mathcal{Q}^{\text{sc}}\ ,\ |q_i| \leq q_i^{\max}\}$ be the space of admissible joint configurations. In this section, we determine the space of admissible parameters

$$\mathcal{A} := \{(\boldsymbol{p}, \lambda) \mid \text{IK}(\boldsymbol{p}, \lambda) \in \mathcal{Q}\}\,. \tag{3.49}$$

Recall the definition of the parameter vector $\boldsymbol{p} := \left[r^{\text{ref}}, \gamma^{\text{ref}}, \beta^{\text{ref}}, \gamma^{\text{ee}}, \beta^{\text{ee}}, \psi^{\text{ee}}\right]^\top$ from (3.20). Only r^{ref} is of linear nature and thus has a limited range. The remaining parameters describe angles and hence need not be limited. While IK_4 directly relates joint limits of the elbow joint with the admissible range of r^{ref}, the null-space parameter λ is related to all remaining joints. Each of which can potentially exclude partitions of the full range of λ. The set of admissible parameters $\mathcal{A}$ must consider all joint limits and results from the intersection

$$\mathcal{A} = \bigcap_{i=1}^{n} \mathcal{A}_i\ , \tag{3.50}$$

of the n individual joint-related portions.

3.3.5.1 Shoulder-Wrist Distance r^{ref}

Elbow joint 4 directly limits the parameter r^{ref}. Solving (3.43d) for r^{ref} gives

$$r^{\text{ref}}(q_4) := \sqrt{(r_{SE})^2 + (r_{EW})^2 - 2r_{SE}r_{EW}\cos(\pi - q_4)} \tag{3.51}$$

and defines the lower and upper bounds

$$r^{\text{ref}}(q_4^{\max}) \ \leq r^{\text{ref}} \leq \ r^{\text{ref}}(0) \tag{3.52}$$

with the upper boundary $r^{\mathrm{ref}}(0)$ being the stretched out configuration of the robot. This defines

$$\mathcal{A}_4 := \left\{ (\boldsymbol{p}, \lambda) \in \mathcal{P} \;\middle|\; \sqrt{(r_{SE})^2 + (r_{EW})^2 - 2 r_{SE} r_{EW} \cos\left(\pi - q_4^{\max}\right)} \leq r^{\mathrm{ref}} \leq r_{SE} + r_{EW} \right\} \tag{3.53}$$

as the admissible parameter space w.r.t. joint 4.

3.3.5.2 Null-Space Parameter λ

All remaining joints, i.e. shoulder joints $\{q_1, q_2, q_3\}$ and wrist joints $\{q_5, q_6, q_7\}$, limit parts of the null-space parameter λ. The 4-quadrant $\operatorname{atan2}(\cdot)$ functions from (3.43), however, are difficult to symbolically rewrite in terms of λ due to there piecewise definition. To circumvent this, we further introduce IK mappings that calculate the absolute joint angles. We define the extraction map of absolute values of the Euler sequence $|\operatorname{eul}_{ZYZ}| : \mathrm{SE}(3) \to \mathbb{R}^3_+$ as

$$|\operatorname{eul}_{ZYZ}| : \quad \mathrm{SE}(3) \to \mathbb{R}^3_+, \quad \boldsymbol{R}_{zyz} \mapsto \begin{bmatrix} \arccos\left(\dfrac{[\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,1)}}{\sin\left(\arccos\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,3)}\right)\right)} \right) \\ \arccos\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,3)}\right) \\ \arccos\left(\dfrac{-[\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(1,3)}}{\sin\left(\arccos\left([\boldsymbol{R}_{zyz}(\boldsymbol{z})]_{(3,3)}\right)\right)} \right) \end{bmatrix} =: \begin{bmatrix} |\gamma| \\ |\beta| \\ |\psi| \end{bmatrix} \tag{3.54}$$

which is used to find the absolute angles of the shoulder and wrist joints

$$\begin{bmatrix} |q_1 + \gamma^{\mathrm{ref}}| \\ |q_2| \\ |q_3| \end{bmatrix} (\boldsymbol{p}, \lambda) := |\operatorname{eul}_{ZYZ}|(\boldsymbol{R}_{3B}(\boldsymbol{p}, \lambda)) \tag{3.55a}$$

$$\begin{bmatrix} |q_5| \\ |q_6| \\ |q_7 + \psi^{\mathrm{ee}}| \end{bmatrix} (\boldsymbol{p}, \lambda) := |\operatorname{eul}_{ZYZ}|(\boldsymbol{R}_{74}(\boldsymbol{p}, \lambda)) \tag{3.55b}$$

analogously to the mapping eul_{ZYZ} from the previous Section 3.3.3. See Section B.3 for the full definition of the absolute valued IK functions. Note that the mappings admit the same symmetrical assignments between the shoulder and wrist portion as the actual IK mapping discussed in Section 3.3.3.5.

Due to the concise formulations of the IK (3.55a), all functions can be solved for the null-space parameter λ. By substituting the joint parameters with their respective limit, closed-form expressions are formed that deliver s_i candidates for lambda ranges

$$\boldsymbol{\lambda}_i^{\max} : \quad \mathbb{R}^7 \times \mathbb{R} \to \mathbb{C}^{s_i}, \quad (\boldsymbol{p}, q_i^{\max}) \mapsto \boldsymbol{\lambda}_i^{\max}(\boldsymbol{p}, q_i^{\max}) \quad \forall i \in [1, 7] \setminus 4 \tag{3.56}$$

according to the $i = [1, 7]$ joints. For q_2 and q_6, the respective middle joints of the shoulder and wrist angle tuples (q_1, q_2, q_3) and (q_5, q_6, q_7), we directly find $s_2 = s_6 = 2$ symmetric solutions for a positive and negative null-space limit. However, solving the remaining mappings from IK (3.55a) for λ, results in more solution candidates. This results from the fact that, depending on the parameter configuration, these joints have the potential for cyclic behavior for a linear increase in λ at a fixed pose (discussed in [SKY+08]). Joints q_3 and q_5 can thus reach up to $s_3 = s_5 = 4$ null-space angles marking a joint limit. The first joint q_1 and last joint q_7 do also offer up to 4 critical values for λ, however, due to additional additive parameters γ^{ref} and ψ^{ee} resp., it is necessary to additionally consider solutions for $|-q_1+\gamma^{\mathrm{ref}}|$ and $|-q_7+\gamma^{\mathrm{ref}}|$. These solutions are evaluated with $\boldsymbol{\lambda}_1^{\max}(\boldsymbol{p}|-\gamma^{\mathrm{ref}}, q_1^{\max})$ and $\boldsymbol{\lambda}_7^{\max}(\boldsymbol{p}|-\psi^{\mathrm{ee}}, q_7^{\max})$. Consequently, $s_1 = s_7 = 8$ solution candidates for the first and last joint of the kinematic have to be considered.

Besides knowing the value of a critical limit, it is further essential for many applications to know if it expresses an upper or a lower limit. Similar to the approach in [FFE+18], the partial derivatives of the null-space range mappings $\boldsymbol{\lambda}_i^{\max}$ w.r.t. the corresponding joint angle limit are used to characterize each limit candidate. For every $\ell \in \boldsymbol{\lambda}_i^{\max}$, the corresponding partial derivative is evaluated to decide

$$\ell \in \begin{cases} \text{is upper limit} & \text{if } \operatorname{sign}(\ell)\,\frac{\partial \boldsymbol{\lambda}_i^{\max}}{\partial q_i^{\max}} > 0 \\ \text{is lower limit} & \text{if } \operatorname{sign}(\ell)\,\frac{\partial \boldsymbol{\lambda}_i^{\max}}{\partial q_i^{\max}} < 0 \\ \text{is no limit} & \text{otherwise.} \end{cases} \tag{3.57}$$

In a second step, all solution candidates in $\boldsymbol{\lambda}_i^{\max}$ are tested for validity, to define the sets of actual upper and lower null-space limit angles

$$\mathcal{L}_i^{\mathrm{up}}(\boldsymbol{p}) := \left\{ \lambda \in \boldsymbol{\lambda}_i^{\max} \;\middle|\; \lambda \in \mathbb{R} \;\wedge\; |\mathrm{IK}_i(\boldsymbol{p},\lambda)| = q_i^{\max} \;\wedge\; \operatorname{sign}(\lambda)\,\frac{\partial \lambda}{\partial q_i^{\max}} > 0 \right\} \quad \forall i \in [1,7] \setminus 4 \tag{3.58a}$$

$$\mathcal{L}_i^{\mathrm{low}}(\boldsymbol{p}) := \left\{ \lambda \in \boldsymbol{\lambda}_i^{\max} \;\middle|\; \lambda \in \mathbb{R} \;\wedge\; |\mathrm{IK}_i(\boldsymbol{p},\lambda)| = q_i^{\max} \;\wedge\; \operatorname{sign}(\lambda)\,\frac{\partial \lambda}{\partial q_i^{\max}} < 0 \right\} \quad \forall i \in [1,7] \setminus 4. \tag{3.58b}$$

These upper and lower limits form j pairwise ranges $\Lambda_{i,j}$ and define the remaining admissible parameter sets

$$\mathcal{A}_i := \left\{ (\boldsymbol{p},\lambda) \in \mathcal{P} \;\middle|\; \lambda \in \bigcup_j \Lambda_{i,j} \right\} \quad \forall i \in [1,7] \setminus 4\,, \tag{3.59}$$

related to shoulder and wrist joints.

The full intersection set $\mathcal{A}$, as defined in (3.50), may consist of several separate regions. Directly evaluating all critical values of λ is especially interesting whenever planning a continuous path in task space. We apply the admissible parameter space in application sections 3.5.1.3 and 3.5.2.2. All full function definitions of the limit candidates $\boldsymbol{\lambda}_i^{\max}$ are summarized in Appendix B.4.

3.4 Results

This section contains an evaluation of the task space manipulability framework developed in this chapter. We first give a run-time comparison to show the computational advantage of our closed-form expression in comparison to general numerical solutions. We show that

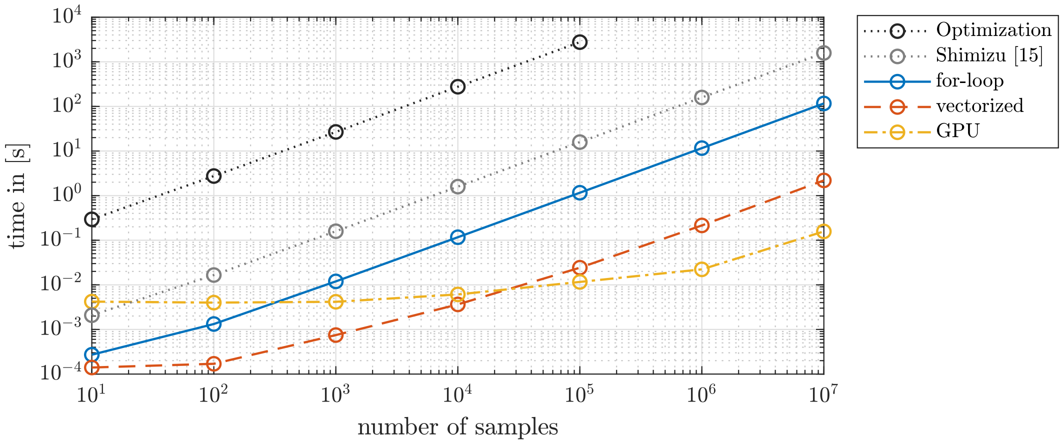


Figure 3.5: Run-time comparison of processing N poses w.r.t. their task space manipulability. Considered are the Matlab robotics IK solver based on nonlinear optimization, the analytical IK solver by Shimizu et al. [SKY$^+$08], and the presented approach in three versions: a conventional sequential loop structure as well as vectorized evaluation on the CPU and GPU.

uniform sampling in the new parameter space results in a superior probability distribution of the manipulability in comparison with direct sampling in joint space. Further, the sensitivity of the manipulability measure w.r.t. the parameters is analyzed.

3.4.1 Accuracy

Unlike numerical IK solvers that approximate the inverse mapping iteratively [GBF85], or CLIK solvers [CT14, Ant09, Bjo18] that converge to the solution from a control point of view, the analytical nature of our closed-form task space manipulability expression delivers exact results in a single iteration.

3.4.2 Run-time Comparison

Complete evaluation of the closed-form IK and M mapping as single expressions allows automatic code generation of the symbolic expressions with e.g. the *MATLAB Coder™* toolbox. These expressions allow array operations, or *vectorization* in Matlab such that a large number of solutions can be evaluated simultaneously. This leads to a significant computational boost, compared to algorithms that rely on matrix arithmetic and consequently have to sequentially evaluate multiple evaluations in programmatic loops. This property makes it further straightforward to calculate the task space manipulability of multiple samples on a powerful graphics processing unit (GPU). Besides different versions of our presented algorithm, we also tested the run-time of [SKY$^+$08], representing typical analytical IK approaches in literature, and the *R5* optimization-based IK algorithm from the *Robotics System Toolbox™* for Matlab representing iterative solver approaches. Figure 3.5 shows a run-time comparison[4] of calculating

[4] The run-time comparison was conducted on with Intel(R) Core(TM) i9-9900X CPU @ 3.50 GHz, 128 GB memory, and a *NVIDIA TITAN V* graphics card.

the manipulability measures

$$\mu_n := (\mathrm{M} \circ \mathrm{IK})(\boldsymbol{p}_n), \quad \text{for } n = [1, N] \tag{3.60}$$

of N random samples $\boldsymbol{p}_n$.

As expected, the iterative optimization algorithm is the computationally most expensive solution method. It required an average of 37 iterations per pose[5] and did not allow for direct selection of the arm angle. Two orders of magnitude faster and in addition producing exact inverse solutions are the analytical IK solvers found in literature. They rely on matrix calculus and thus a for-loop structure for evaluation of multiple poses.

Our approach, which is entirely reduced to direct individual expressions, is over 10 times faster when implemented with the same conventional for-loop structure. Already for 200 evaluated samples, a simultaneous vectorized evaluation achieves another performance increase of factor 10. At the maximal evaluated amount of 10^7 samples, vectorization enables an even 50 times faster computation, compared to the implementation using for-loops. The advantage of calculating the task space manipulability on a GPU starts at an amount of 10^5 sample points. For a smaller number of samples, the overhead of initializing the data on the GPU does not pay off. Processing 10^7 samples, calculations on the GPU are 10 times faster then vectorized treatment on the CPU, and even 700 times faster than for conventional loop structures. Note that all time measurements include the generation of random samples on the CPU and GPU respectively.

Considering real-time application for a robot with a typical 1 kHz sampling rate, our approach allows evaluation of 1000 end-effector poses for their task space manipulability.

3.4.3 Sampling in Task Space

Not having to calculate the IK in an iterative fashion as done by CLIK solvers, evaluating manipulability directly in task space is computationally not much more expensive than directly calculating manipulability in joint space. However, choosing a different space for sampling random poses do have an influence on the probability distribution of resulting manipulability measures.

Before analyzing this difference, we first discuss the used sampling strategies. For a fair comparison, we cover the entire space without consideration of possible limits on the individual joints or parameters.

Let $u \in \mathbb{R}$ be a random number drawn from a uniform distribution in the range of $[0, 1]$. Uniform sampling in joint space is straightforward with

$$q_i^{\text{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto -\pi + 2\pi u \qquad \forall i \in [1, 7] \tag{3.61}$$

due to the independence of its joints $\boldsymbol{q} \in \mathbb{R}^7$.

[5] Optimization Algorithm: BFGS Gradient Projection; solution tolerance: 0.01;

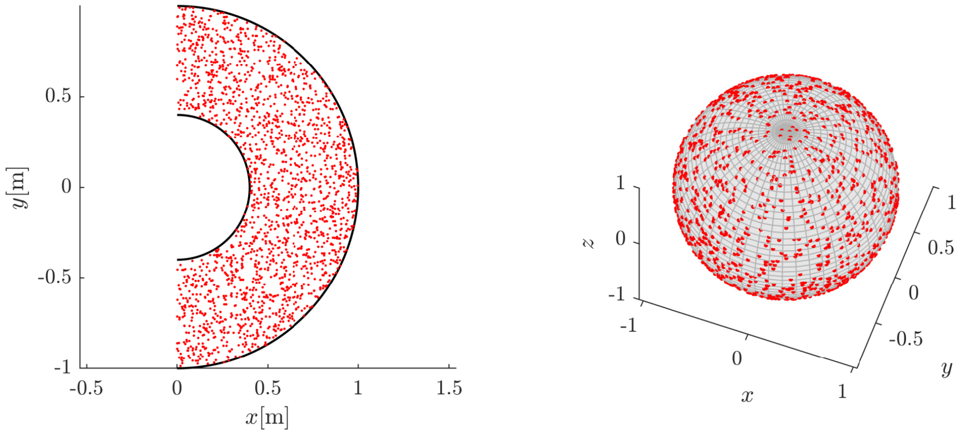


Figure 3.6: Uniform distributed sampling of the task space (2000 samples).

For a random end-effector pose sample $(\boldsymbol{p}^{\mathrm{red}}, \lambda) = [r^{\mathrm{ref}}, \beta^{\mathrm{ref}}, \gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \lambda]^\top$ from the parameter space, one can choose the same strategy

$$p_1^{\mathrm{naive}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto r_{\mathrm{min}}^{\mathrm{ref}} + (r_{\mathrm{max}}^{\mathrm{ref}} - r_{\mathrm{min}}^{\mathrm{ref}})u \tag{3.62a}$$

$$p_i^{\mathrm{naive}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto -\pi + 2\pi u \qquad \forall i \in [2,5] \tag{3.62b}$$

with respective scaling for the linear parameter r^{ref}. However, this *naive* form of sampling does not lead to a uniform distribution of samples in the task space SE(3), due to the interdependence of the coordinate components.

Recall that the first two parameters r^{ref} and β^{ref} describe translation in polar coordinates. Unlike in Cartesian coordinates, the base vectors are not constant. Consequently, direct uniform sampling of the radial coordinate r^{ref}, leads to sparser sampling further from the origin, due to the increasing circumference proportionally to r^{ref}. Proper uniform sampling of the translational part can be achieved by

$$p_1^{\mathrm{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto \sqrt{\left(r_{\mathrm{min}}^{\mathrm{ref}}\right)^2 + \left(\left(r_{\mathrm{max}}^{\mathrm{ref}}\right)^2 - \left(r_{\mathrm{min}}^{\mathrm{ref}}\right)^2\right) u} \tag{3.63a}$$

$$p_2^{\mathrm{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto -\pi + 2\pi u. \tag{3.63b}$$

An efficient method of uniform sampling on SO(3), i.e. 3D orientations, is proposed by Kuffner [Kuf04]. Uniform sampling of the individual angles of the Euler sequence results in a bias towards the polar regions of the unit sphere. He proposes to use an arctan function on the second angle to compensate for this bias. Uniform sampling of the end-effector orientation, parametrized by γ^{ee} and β^{ee}, is thus achieved with

$$p_3^{\mathrm{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto -\pi + 2\pi u \tag{3.63c}$$

$$p_4^{\mathrm{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto \arccos\left(1 - 2u\right). \tag{3.63d}$$

The last portion in our parameter tuple $(\boldsymbol{p}, \lambda)$ is the null-space parameter λ that is independent and thus remains

$$p_5^{\mathrm{uniform}}: \quad \mathbb{R} \to \mathbb{R}, \quad u \mapsto -\pi + 2\pi u. \tag{3.63e}$$

Figure 3.6 illustrates the uniform sampling of the task space applying the uniform sampling strategy (3.63).

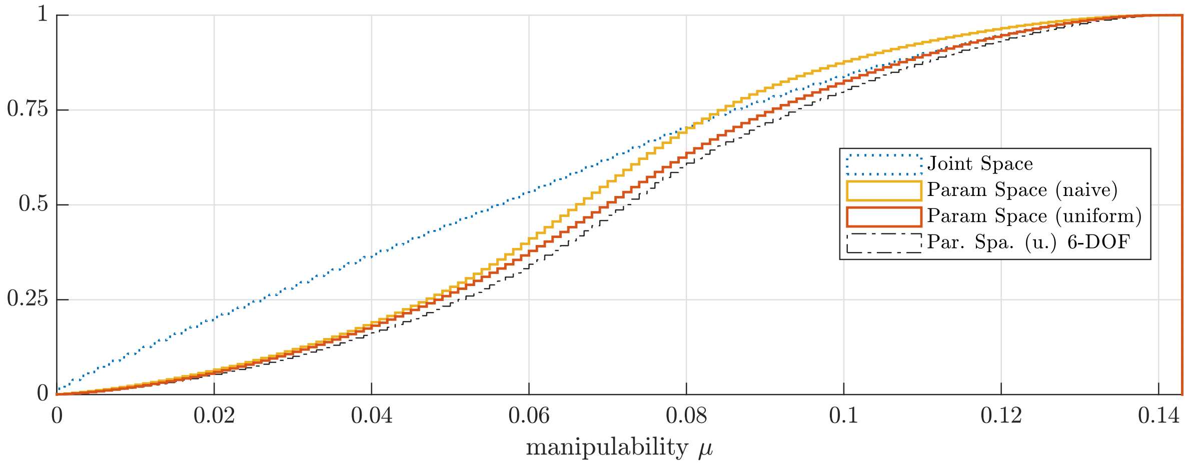


Figure 3.7: Approximated cumulative distribution function (CDF) from a histogram of manipulability w.r.t. different sampling strategies (10^7 samples).

The above-discussed sampling strategies are now analyzed in conjunction with their respective mapping to the manipulability measure. Figure 3.7 shows the approximated CDF of manipulability resulting from 10^7 random samples. It shows that random sampling in joint space according to (3.61) is more likely to result in a joint configuration with poor manipulability of the robot. Uniform sampling in parameter space (3.63) produces much fewer joint configurations with poor manipulability, while at the same time more configurations with high manipulability. Naive sampling in parameter space (3.61) performs similarly good in the low manipulability section. However, it produces also fewer configurations with high manipulability. Considering a conventional 6-DOF robot, i.e. fixing the null-space parameter λ to 0 or π, results in a slightly better probability density function (PDF) than for the discussed 7-DOF mechanism. This is a surprising result, as it is always argued that the redundancy improves manipulability. While it is true that the additional DOF has the potential to improve performance measures, poor exploitation might achieve the opposite. Kuhlemann et al. [KJES16] showed in different use-cases, that the seventh DOF of the *KUKA LBR iiwa* increased the average dexterity by 16 % in comparison to a conventional 6 DOF *KUKA KR 10*. Both, the shortcomings of the naive parameter sampling strategy, and the apparent advantage of the 6-DOF mechanism are discussed in Section 3.4.4.4.

The average normalized manipulabilities achieved, are 37 % for uniform joint space sampling, 43 % for naive parameter space sampling, and 50 % for uniform sampling in parameter space. All numbers are w.r.t. the maximal encountered manipulability.

3.4.4 Parameter Sensitivity Analysis of Manipulability in Parameter Space

The sensitivity of the task space manipulability w.r.t. its parameters are analyzed by generating 10^7 random samples according to (3.63). These samples represent a uniform distribution of task space configurations. Figure 3.8 shows the bi-variate histograms of manipulability $\mu(\boldsymbol{p}^{\mathrm{red}}, \lambda)$ w.r.t. to the individual parameters.

Colors approximate the PDF of $\mu(r^{\mathrm{ref}}, \beta^{\mathrm{ref}}, \gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, \lambda)$ at fixed values of the respective pa-

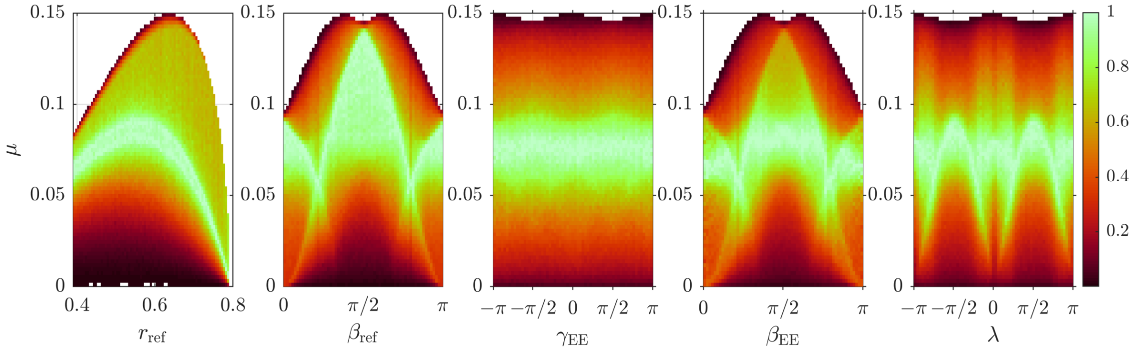


Figure 3.8: Bi-variate histograms of $\mu(r^{\text{ref}}, \beta^{\text{ref}}, \gamma^{\text{ee}}, \beta^{\text{ee}}, \lambda)$ w.r.t. to the individual parameters, based on 10^7 uniformly distributed parameter space samples. Colors are normalized along with the particular value of the parameter on the x-axis.

rameter. For all parameter values, we find unimodal[6] distributions.

3.4.4.1 Translation Parameters r^{ref} and β^{ref}

The PDF of μ along the shoulder-wrist distance r^{ref} shows a preferred value of 0.57 m. Although a manipulability optimizing configuration cannot be found at this given value, the mode[7] of the corresponding PDF has the highest value of manipulability. Further, the probability of good manipulation is decreasing with r^{ref} towards the *workspace singularity*, i.e. a fully stretched arm of robot configuration.

The polar angle β^{ref} between the vertical and the shoulder-wrist reference vector has the highest manipulability mode at $\frac{\pi}{2}$ rad, although manipulability maximizing configurations are not found. For values approaching 0 and π rad, i.e. placing the wrist in line with the axis of base joint q_1, typically cause so-called *shoulder singularities* on conventional 6-DOF robots. While the 7-DOF kinematics does not necessarily result in a kinematic singularity, high manipulability is neither possible.

3.4.4.2 Orientation Parameters γ^{ee} and β^{ee}

The third parameter γ^{ee}, which describes a rotation around the shoulder-wrist vector, is the only one that seems to cause little variation in the manipulability PDF and does not allow a conclusion over a preferred configuration.

The consecutive rotation angle β^{ee} shows a similar influence as the reference angle β^{ref}. However, the mode of these PDFs is less prominent and tendentiously marks a lower manipulability.

[6] unimodal: single maxima in PDF
[7] mode: local maxima in PDF

3.4.4.3 Null-Space Parameter λ

The null-space parameter λ reveals that the highest manipulabilities can be found at $\lambda = \{0, \pm\pi\}$rad, i.e. the conventional upper and lower elbow configuration of 6-DOF kinematics. Although missing the absolute top manipulability poses, only small deviations of about ±0.1 rad from these configurations result in a decrease of the manipulability mode of 25 %, i.e. from 0.08 to 0.06. Better modes are found at $\lambda = \{\pm\frac{\pi}{2}\}$rad. Not only is their peak at a slightly higher manipulability of 0.85, but they are also less sensitive to a parameter change in λ. The latter is especially valuable for staying agile during unforeseen events.

3.4.4.4 Discussion of Manipulability in Different Sampling Strategies

The different sampling strategies discussed in Section 3.4.3 result in differences in the approximated CDFs, cf. Figure 3.7.

Naive vs uniformly distributed sampling The difference between naive and uniform sampling solely affects parameters r^{ref} and β^{ee}. That is, the corresponding uniform sampling functions (3.63a) and (3.63d) correct the biases of the radial coordinate r^{ref} towards the origin, and the orientation towards the pole regions with azimuthal angle $\beta^{\text{ee}} = \{0, \pi\}$rad, respectively. Consequently, these regions are sparser sampled in the uniformly distributed strategy. While this correction is negligible for the range of r^{ref} in this particular robot example, the improvement of the CDF towards better manipulability stems from a sparser sampling of the boundary regions of β^{ee}. Because exactly these boundaries lack high manipulability poses, as visible in the according bi-variate histogram in Figure 3.8.

6-DOF vs 7-DOF kinematics According to Section 3.4.4.3, the apparent slight advantage of uniform distributed sampling of a conventional 6-DOF robot only holds for the over-all manipulability distribution illustrated in Figure 3.7. The parameter-specific histogram w.r.t. the arm angle λ in Figure 3.8, on the other hand, reveals that the conventional 6-DOF configurations $\lambda = \{0, \pm\pi\}$rad do have a good manipulability distribution, but $\lambda = \{\pm\frac{\pi}{2}\}$rad are preferable. A 7-DOF kinematics hence not only enables agile adaptation of the kinematic structure, but also contains arm angles that have a better PDFs of manipulability than its 6-DOF counterpart. At the same time, other arm angles show higher variability in the histogram and are more prone to decrease performance. An increase in manipulability by the additional DOF thus relies on a well-conceived utilization of such.

3.4.5 Number of Local Optima

While the analysis shown in the previous section gives insight in the probability distribution of the manipulability measure, it does not allow conclusions on how manipulability changes along the null-space. Table 3.1 lists the number of local optima for a given end-effector pose. It shows that 80 % of the robot poses do not have a unique manipulability maximizing null-space solution, but up to 4 distinct optima.

number of optima	1	2	3	4
percentage	20%	41%	27%	12%

Table 3.1: Distribution of local optima among 10^7 samples.

3.5 Applications

Two application directions that benefit from the closed-form expressions of the task space manipulability are outlined in this section. First, we demonstrate how global optimization problems can be formulated, that profit from massive multi-start point pre-evaluation. Second, we propose a novel way of real-time redundancy resolution on position level, that enables global manipulability optimization of single poses as well as for provided end-effector trajectories in SE(3).

3.5.1 Optimal Robot Placement

The analytic results from the previous Section 3.3, allow formulating interesting questions in terms of optimization problems. We consider the problem of optimal placement of the robot.

3.5.1.1 Best Overall Robot Configuration

The most basic optimization problem we considered is the question of finding the best overall robot configuration w.r.t. manipulability. Mathematically, this problem can be stated as an unconstrained optimization problem

$$\underset{\boldsymbol{q}}{\text{maximize}} \quad \mu(\boldsymbol{q}) \tag{3.64}$$

directly finding the optimal joint configuration w.r.t. the manipulability measure. The global optima is found with a multi-start strategy [Dix78], where random samples are drawn from the admissible parameter space $\mathcal{P}$ and used as starting points for local optimizations. Figure 3.9, left side, shows the results of such a global optimization process with 1000 starting points. Note that the same problem can be formulated in *parameter space* and does yield the same result. All optimization iterations result in one of 8 equally good global optima, which can be reduced to 4 solutions due to symmetry of the shoulder joint. They further describe configurations in the pure xz-plane with $\lambda \in \{0, \pm 180\}°$. This is equivalent to the configurations achievable by a conventional 6-DOF robot.

3.5.1.2 Best Robot Configuration for Multiple Task Poses

In industrial settings, robots are often required to work at a certain number $i \in \mathbb{N}_+$ of different task poses $\boldsymbol{z}_i$. While the relative distances $\Delta\boldsymbol{z}_i = \boldsymbol{z}_i - \boldsymbol{z}_1$ between this poses is defined, the optimal placement of the robot can be found by solving the optimization problem

$$\underset{\boldsymbol{z},\lambda}{\text{maximize}} \quad \sum_i (\mathrm{M} \circ \mathrm{IK} \circ \mathrm{TSP})(\boldsymbol{z} + \Delta\boldsymbol{z}_i, \lambda) \tag{3.65}$$

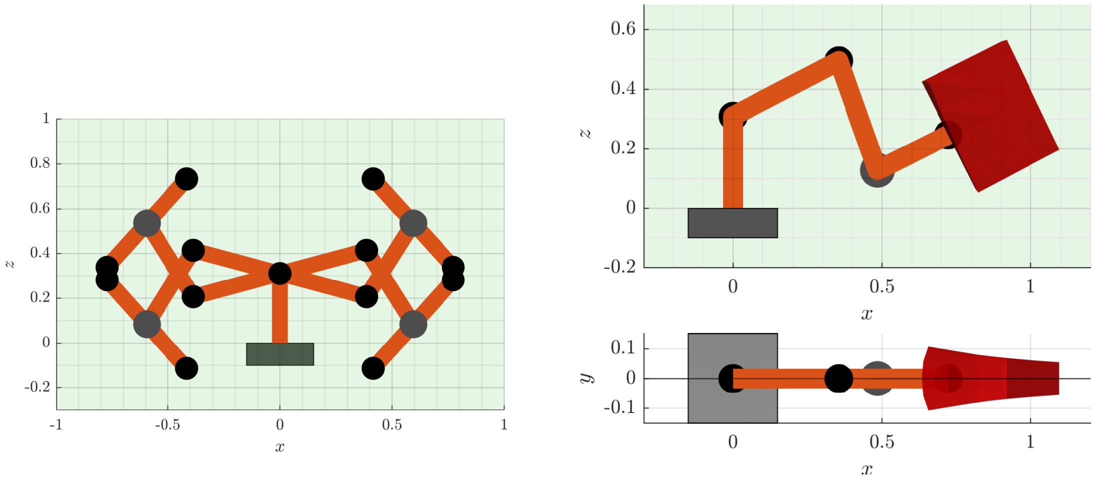


Figure 3.9: Results of the task space manipulability optimization of a robot mounting pose.

to find the relative pose $\boldsymbol{z}$ that maximizes the average manipulability of all i poses. Solving this problem directly, results in an infinite number of global poses. These solutions are rotationally symmetric around the base joint q_1 as well as the last joint q_7, as both these joints do not have an influence on the manipulability of the 7-DOF robot structure at consideration (discussed in Section 3.3.1).

The complexity of the optimization problem, as well as the number of global optima, can be drastically reduced by formulating the same problem in the lower dimensional parameter space

$$\underset{\boldsymbol{p},\lambda}{\text{maximize}} \quad \sum_i (\mathrm{M} \circ \mathrm{IK})(\boldsymbol{p} + \varDelta\boldsymbol{p}_i, \lambda) \tag{3.66}$$

where $\boldsymbol{p}_i = \mathrm{TSP}(\boldsymbol{z}_i)$. The resulting optimal $\boldsymbol{p}$ can eventually be mapped to the corresponding task space parameter $\boldsymbol{z} = \mathrm{TSS}(\boldsymbol{p})$. This result is useful for deciding on how to mount a robot relative to a given set of task poses $\boldsymbol{z}_i$, or recalculating it online if task poses are time-variant and the robot structure is e.g. mounted on a mobile platform.

3.5.1.3 Optimizing Robot Mounting Positions Regarding a Workspace Envelope

In a modern scenario where robots are not only expected to repetitively execute the same tasks, a set of pre-defined task poses cannot always be formulated. But it is rather necessary for the robot to perform well in a defined workspace volume, e.g. given as a cubical volume $V = [-\varDelta\frac{x}{2}, +\varDelta\frac{x}{2}] \times [-\varDelta\frac{y}{2}, +\varDelta\frac{y}{2}] \times [0, +\varDelta z]$. Due to all mappings involved in the task space manipulability being continuous, formulating a cost function for such a volume can be done using Fubini's theorem [Fub07]. It allows calculation of the volume integral as triple integral. The objective for this optimization problem in task space reads

$$\begin{aligned} &\underset{\boldsymbol{z}_0,\lambda}{\text{maximize}} \iiint\limits_V (\mathrm{M} \circ \mathrm{IK} \circ \mathrm{TSP})(\boldsymbol{z}_0 + \boldsymbol{z}(x,y,z), \lambda)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ &\text{subject to} \qquad \mathrm{TSP}\left(\boldsymbol{z}_0, +\boldsymbol{z}(x,y,z), \lambda\right) \in \mathcal{A}, \end{aligned} \tag{3.67}$$

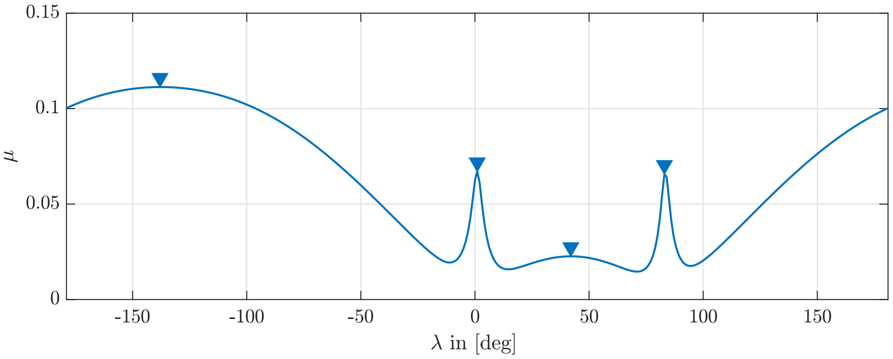


Figure 3.10: Multiple local optima of manipulability μ in the null-space of $\boldsymbol{p}^{\text{red}} = [0.6, 0.7, 1.4, 0.7]^\top$.

where the optimal task space volume origin $\boldsymbol{z}_0$ needs to be found. This optimization can again be transformed to the lower dimensional parameter space

$$\begin{aligned} \underset{\boldsymbol{p}_0,\lambda}{\text{maximize}} \quad & \iiint\limits_V (\mathrm{M} \circ \mathrm{IK} \circ \mathrm{TSP})(\mathrm{TSS}(\boldsymbol{p}_0) + \boldsymbol{z}(x,y,z), \lambda)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ \text{subject to} \quad & \mathrm{TSP}\left(\mathrm{TSS}(\boldsymbol{p}_0, \lambda) + \boldsymbol{z}(x,y,z), \lambda\right) \in \mathcal{A} \end{aligned} \tag{3.68}$$

with the condition that the whole Volume projected to parameter space must be within the set of admissible parameters. Figure 3.9, right side, shows the result of such a global optimization.

3.5.2 Redundancy Resolution

Solving for optimal robot poses online is essential for a robot to stay agile at all times. We demonstrate how the task space manipulability expressions developed in this chapter can be applied for real-time global manipulability optimization of single poses as well as full trajectories.[8] Note that the trajectory optimization in this chapter only gives a first outlook. An in-depth discussion follows in Chapter 4.

3.5.2.1 Redundancy Resolution for Global Manipulability Optima

Approaches typically found in the literature, focus on local optimization of manipulability based on local gradient information. Analysis of the number of existing local optima from Section 3.4.5, however, revealed that only 20 % of end-effector poses have a unique global optimum. The computational advantage of our approach permits evaluating the manipulability of many poses simultaneously. Given a current robot pose $\boldsymbol{z}$, our framework makes it possible to not only locally improve manipulability, but solve

$$\underset{\lambda}{\arg\max} \quad (\mathrm{M} \circ \mathrm{IK} \circ \mathrm{TSP})(\boldsymbol{z}, \lambda) \tag{3.69}$$

[8] Run-time evaluation conducted in MATLAB 2019a, on a computer with Intel(R) Core(TM) i3-7100 CPU @ 3.9 GHz and 32 GB memory.

with a representative number of null-space solution at a high resolution in real-time. Given the information of this *greedy optimization* strategy, the close-to global optimum configuration can simply be picked. Solving for global optima in 0.25 ms at a resolution of 1° for λ, enables application at typical robot sampling rates of 1 kHz.

Figure 3.10 shows manipulability of the full null-space at a particular configuration. This is an example of a pose with 4 local optima. If the current configuration of the robot is the solution for the given pose with the null-space parameter $\lambda \in [0, 85]^\circ$, a local optimization will only drive the redundancy resolution into a sub-optimal minima. Whereas our approach allows finding the globally best configuration w.r.t. the admissible parameter space.

3.5.2.2 Optimizing Null-Space Solution of Given End-Effector Trajectory

Several approaches can be found in literature, that maximize either the volume of a manipulability ellipsoid [Chi90, Chi87, GKNK06, SRG$^+$16] or a predefined shape of the ellipsoid [JRCC18]. Yet all these approaches consider only local optimization. Finding the best joint configuration for a given pose in task space simplifies to a 1D line search. However, given a full path in SE(3) it is also possible to find an optimal elbow trajectory that maximizes e.g. the average manipulability while avoiding getting trapped in regions of poor manipulability. Note that a real manipulation task relies on a sophisticated path planner, capable of generating task-related paths that avoid obstacles while potentially fulfilling additional criteria. Knowledge about the task space manipulability, e.g. provided by our approach, may even be exploited by such a planner. This is, however, not the direct scope of this thesis. Instead, for a minimal working example in this chapter, we use direct interpolation

$$\boldsymbol{p}(s) = s\boldsymbol{p}^{\text{start}} + (s-1)\boldsymbol{p}^{\text{end}} \quad \text{with} \quad s = [0, 1] \tag{3.70}$$

between two poses as a simple path planner. Given are two random poses as depicted in Figure 3.11 to the left. On the right side of Figure 3.11, a contour plot of the manipulability of the full null-space along the trajectory is shown. Red lines indicate not passable values in the null-space due to joint limits, cf. Section 3.3.5. The blue line marks the trajectory that results from local optimization of manipulability. Note that at $s = 0.4$ the local optimization hits a joint limit of q_2. We stopped the line here, because it depends on a potential strategy for joint limit avoidance, which is not the scope of this chapter. A global optimization strategy that has predictive knowledge of the full null-space development, can exploit an initially sub-optimal path toward negative values of λ, to circumvent the region of poor manipulability between $s = [0.6, 1]$. But this is usually not feasible in an online scenario with conventional global optimization strategies.

The computational advantage of our strategy, as seen in Figure 3.5, allows the computation of such a map with e.g. a resolution of 100 steps in both parameters, s and λ, in under 5 ms. In combination with an online trajectory generator directly on SE(3) e.g. Chapter 2, this qualifies our task space manipulability approach to be used for predictive online manipulability optimization e.g. with a receding horizon.

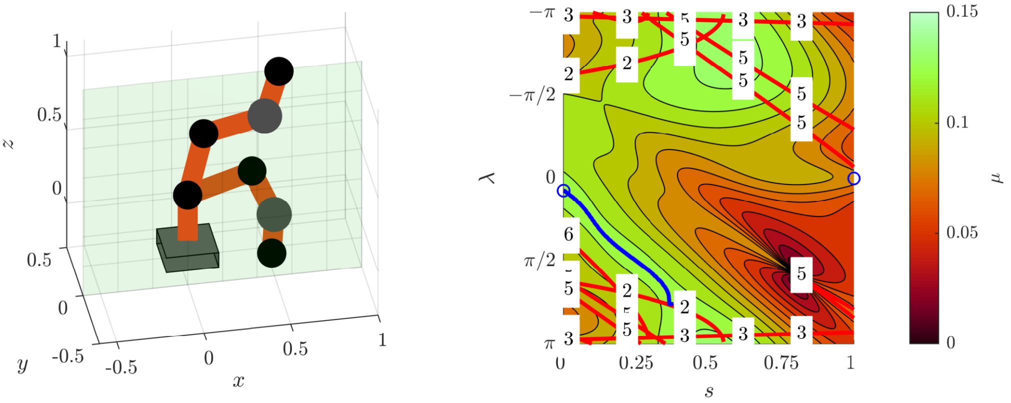


Figure 3.11: Null-space manipulability over a parameter trajectory. The left image shows an exemplary start $\boldsymbol{p}(s = 0)$ and end configuration $\boldsymbol{p}(s = 1)$. The right contour plot shows the manipulability $\mu(\boldsymbol{p}, \lambda)$ of the full null-space. Red lines mark the limits $\boldsymbol{\lambda}_i^{\max}(\boldsymbol{p}, q_{\max})$ of the admissible null-space region. The numbers refer to the invoking joint i. Blue circles mark desired $\lambda^{\mathrm{des}}(s = 0)$ and $\lambda^{\mathrm{des}}(s = 1)$, and the blue line marks the trajectory that locally optimizes μ.

3.6 Chapter Conclusion

Today’s demand for adaptive and reactive robot behavior requires to sustain agility of a kinematic structure at all times. While manipulability is a common metric in robot research to quantify the capabilities of a robot at a given joint configuration, the robot task is directly defined in end-effector poses, that allow for multiple possible solutions. Unlike common metrics, which do not include the robot IK, a task space manipulability formulation is required to directly map an end-effector pose together with its null-space solution onto the manipulability metric.

To achieve reactive robot behavior, optimization of the null-space at given poses must be performed online. In general, this requires efficient evaluation of a large number of configurations, especially in the case of redundant robots. In this chapter we developed a new closed-form approach for calculating manipulability directly from task space poses, for a redundant 7-DOF S-R-S serial robot kinematics. A novel parametrization of the task- and null-space leads to concise IK as well as admissible parameter mappings, which show symmetry in the structures of their individual expressions. Analysis of the resulting task space manipulability further revealed that the majority of end-effector poses do not have a unique, manipulability maximizing null-space solution. We thus argue that local optimization of the manipulability measure is not sufficient. A global optimization at high sampling frequencies, however, is not feasible with current approaches in literature. The entire composition of the task space manipulability map proposed in this chapter allows for efficient array operations that can be exploited in vector-optimized programming languages as well as GPU computing. Consequently, the simultaneous computation of a large number of poses in real-time is made possible. Our method, therefore, enables global online optimization of manipulability for single poses and even full SE(3) trajectories.

Combining our task space manipulability approach with online planners opens an interesting field of predictive redundancy resolution for global manipulability optimization. This idea is further elaborated in the next chapter. Future work will focus on further application development of our framework.

"Science can amuse and fascinate us all, but it is engineering that changes the world."

— Isaac Asimov (1920–1992)

4 Globally Optimal Online Redundancy Resolution for Serial 7-DOF Kinematics Along End-Effector Trajectories

This chapter was previously published in [HW21].

Redundant robots offer the possibility of improving agility, compared to their non-redundant counterparts, by exploiting the additional kinematic DOFs to increase a measure called *manipulability*. While it is common to maximize the manipulability measure during redundancy resolution locally, global optimization of a full trajectory is usually computationally too expensive and thus only considered for offline procedures in current literature. However, local maximization is prone to be sub-optimal and at times even fails at preserving agility of a robot that ought to be reactive. In this chapter we build upon our previous contributions on *online trajectory generation on SE(3)* and *closed-form task space manipulability of a 7-DOF serial robot*, and combine it with graph search techniques for global optimization. This enables, for the first time, online trajectory generation with globally optimal redundancy resolution regarding manipulability, to maintain agility in reactive robot behavior.

4.1 Introduction

We argued in Chapter 2 that for *safety*- and *human comfort* aspects in HRC scenarios, it is necessary to generate trajectories online and directly for the end-effector movement in the SE(3) rather than in joint space. While the proposed OTG on SE(3) is independent of a potential robot kinematics guiding the end-effector, a mapping onto joint trajectories that are eventually executed is inevitable. Continuing our focus on the reaction capacity of the robot, we developed analytic closed-form expressions for the most commonly used articulated 6- and 7-DOF serial robot kinematics in Chapter 3. Although redundant kinematics have the potential of improving the movement capability of a robot, an effective exploitation of this potential relies on a proper redundancy resolution strategy. It is common practice to therefore solve the redundancy by maximizing metrics such as Yoshikawa's manipulability measure [Yos85b].

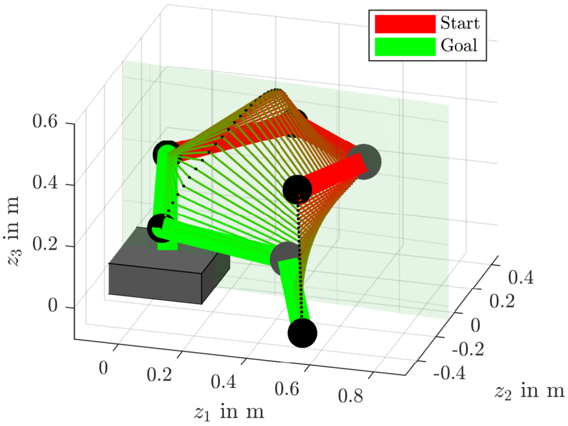


Figure 4.1: Straight line end-effector movement with change in orientation from start (red) to goal configuration (green), with equitemporal in-between configurations of the globally optimal null-space solution trajectory indicated by the color gradient.

Based on our previous work on an OTG on SE(3) and closed-form expressions for Yoshikawa's manipulability measure directly in the same SE(3) task space, cf. chapters 2 and 3, this chapter exploits the computational efficiency of these methods to enable – for the first time in literature – an online capable redundancy resolution method for globally optimal solutions w.r.t. manipulability along an end-effector trajectory, cf. Figure 4.1 for an illustration.

4.1.1 Related Work

In this brief literature review, we highlight common strategies for local as well as global optimization of different criteria. While the initial research interest from the late 80s till mid 90s yielded in various analytic-centered methods, the last 10 years show a revival with a focus on computationally demanding optimization-based strategies.

Local Pose Optimization Redundancy resolution in general is traditionally treated as a control problem on either velocity or acceleration level, with the correct forward kinematics being the main objective and different performance measures as sub-goals. Khatib [Kha87] proposed a prominent framework for torque-based redundancy resolution. He uses an inertia-weighted pseudoinverse of the Jacobian matrix, which dynamically decouples the task space from the null-space motions. A torque optimizing redundancy resolution strategy was presented by Hollerbach and Suh [HS87]. They compare the unweighted as well as torque-range-weighted null-space acceleration vector algorithms with an inertia-weighted pseudoinverse algorithm, and point out that these null-space based kinetic algorithms yield smaller torques locally, but tend to be unstable for longer trajectories. However, they also mention that even the kinematics pseudoinverse-based method shows instabilities on rare occasions. Sciavicco and Siciliano [SS88] proposed a task space augmentation approach instead. However, this strategy introduces *algorithmic singularities* in addition to the always present kinematic singularities. They occur whenever an augmented constraint task conflicts with the main end-effector task. An extensive review of early works on local redundancy resolution through local optimization before 1989, was published by Nenchev [Nen89]. Chiaverini [Chi97] proposed a singularity-robust task-priority redundancy resolution technique, that can also handle the above mentioned algorithmic singularities. A comparison of different task space control strategies with redundancy resolution in an experimental setup, was conducted by Nakanishi et al. [NCM+05].

Because manipulability measures itself are highly nonlinear functions, locally optimizing such criterion results often in merely local optima. In Chapter 3 we showed that in the case of a serial 7-DOF kinematics, up to 4 local optima are possible and only 20 % of all possible joint configurations admit a unique optimum.

Global Pose Optimization Hauser and Emmons [HE18] developed an algorithm that applies a sampling-based approximation technique using probabilistic roadmaps, to approximate a pseudo-inverse map for global redundancy resolution. The resulting inverse map assigns a unique joint configuration to every end-effector pose in task space, and thus leads to *cyclic solutions*, which are of special interest in industrial contexts. This term refers to solutions that guarantee the same joint configurations for repeated cyclic motions, sometimes also referred to as *conservative motion*. Jin et al. [JLLL17] reformulated the nonlinear local manipulability optimization problem as a convex quadratic program, and apply a dual neural network framework to solve it online.

But only globally optimizing the current pose w.r.t. the manipulability measure may be not sufficient, e.g. for cases where null-space configurations along an end-effector trajectory violate given constraints. In such cases, if the global optima jumps into a region without valid transition or in the worst case inevitably leads to a kinematic singularity, it is necessary to utilize a prediction of the emerging situation in the null-space and consider the full trajectory for optimization. Only then is it possible to initiate a necessary transition with foresight.

Global Trajectory Optimization Nakamura and Hanafusa [NH87] applied Pontryagin's maximum principle to solve redundancy w.r.t. a general performance index of the integral type, and reduce the computation to a minimal value search problem in a space of as many dimensions as the degrees of redundancy for kinematics, and twice the dimension if dynamics are considered. Kazerounian and Wang [KW88] formally showed by using calculus of variations, that a local minimization of the joint accelerations causes a global minimization of joint velocities. Consequently, this results in the same joint trajectories as the approach [NH87], but admits simpler expressions and is more suitable for physical interpretations as well as numerical and symbolic manipulations. An approach tailored to cyclic solutions, was proposed by Choi et al. [CWC92]. None of these global optimization approaches, however, considers manipulability in the optimization criteria for retaining the capability of the robot kinematics to react to unforeseen events.

More recently in literature, some authors also suggested to tackle the global trajectory optimization problem by discretized approximations, to overcome the limitations of analytic formulations. Because the latter typically result in two-point boundary value problems that are complex to solve. While some early works already considered dynamic programming for trajectory generation [SM86, SL87], more recent literature showed increasing interest in this technique again. Guigue et al. [GALH10] demonstrated that dynamic programming inspired approaches bring more flexibility and easily allow for multiple objective functions, guaranteeing Pareto-optimality. Dolgui and Pashkevich [DP09] use graph search for off-line programming of laser cutting robotic systems. They also explicitly incorporate verification of velocity and acceleration constraints. Another graph search approach was proposed by Gao et al. [GPC17]. They apply spline-based post-processing to generate motions of a redundant system for fiber placement in automated manufacturing of composite-based components. Ferrentino and Chiacchio [FC20] give a topological analysis of the problem and formulate a multi-grid search algorithm that even allows for kinematics reconfiguration of the manipulator in the solution trajectories. Unfortunately, this kind of discretized global optimizations are usually computationally very demanding and are thus not suitable for an OTG scenario where response time is safety-critical.

The current literature does not offer an online-capable approach for exploiting kinematic redundancy for global optimization of a manipulability measure along a trajectory.

4.1.2 Contribution

In this chapter we propose a framework that provides globally optimal redundancy resolution trajectories for a 7-DOF robot, based on full trajectories of the end-effector provided in the SE(3) task space. For the first time, this all is achieved at 50 Hz under practically realistic parameters, including the end-effector trajectory generation, and thus qualifies our approach for online application. We further outline how the strategy can be incorporated in an overall SE(3) OTG architecture for reactive robot behavior.

4.2 Problem Description

Given are a current end-effector state $\boldsymbol{x}^{\mathrm{cur}}$ and a desired goal state $\boldsymbol{x}^{\mathrm{des}}$, both consisting of state tuples

$$\boldsymbol{x} := (\boldsymbol{z}, \dot{\boldsymbol{z}}) \tag{4.1}$$

containing the end-effector pose $\boldsymbol{z} \in \mathrm{SE}(3)$ and velocities $\dot{\boldsymbol{z}} \in \mathbb{R}^6$, together with a prediction time horizon $T^{\mathrm{pred}} \in \mathbb{R}_+$. An OTG is applied to generate a time trajectory of the end-effector pose $\boldsymbol{z}(t)$, that drives the end-effector from $\boldsymbol{x}^{\mathrm{cur}}$ to $\boldsymbol{x}^{\mathrm{des}}$. Consider a serial 7-DOF robot kinematics (as illustrated in Figure 4.1) is used to drive the end-effector toward the goal state. It is parametrized in the joint angles $\boldsymbol{q} \in \mathbb{R}^7$. The remaining 1D null-space of the redundant robot kinematics is parametrized via the arm angle $\lambda \in [-\pi, +\pi]$, i.e. the rotation angle of the elbow (joint 4) from an upright position in the xz-plane for $\lambda = 0$, around the axis pointing from the shoulder (joint 2) to the wrist (joint 6). See Section 3.3.2 for a detailed explanation of this parametrization.

To retain maximal agility for reactive robot behavior along $\boldsymbol{z}(t)$, a solution for the arm angle trajectory $\lambda(t)$ has to be found, that globally optimizes the manipulability measure μ

from [Yos85b] in terms of the cost function

$$\min_{\lambda(t)} \int_0^{T^{\text{pred}}} \mu(\boldsymbol{z}(t), \lambda(t)) \mathrm{d}t, \tag{4.2}$$

subject to the set of constraints

$$\mathcal{C} := \{\mathcal{C}^{\text{OTG}}, \boldsymbol{q}^{\max}, \omega_\lambda^{\max}\} \tag{4.3}$$

containing OTG-specific limits as well as limits for robot joint values $\boldsymbol{q}^{\max}$ and the maximal angular velocity of the arm angle $\omega_\lambda^{\max} := \left(\frac{\mathrm{d}\lambda}{\mathrm{d}t}\right)^{\max}$. For reactive robot behavior it is further required, that the algorithm shows online capability.

4.3 Approach

The approach is split up into three distinct parts, as listed in Algorithm 4.1:

1. Given a current and desired state in task space, together with the prediction horizon T^{pred}, an end-effector state trajectory $\boldsymbol{x}(t)$ is predicted for $0 \leq t \leq T^{\text{pred}}$.
2. At $N_t \in \mathbb{N}_+$ equi-temporal discrete poses $\boldsymbol{z}$, the manipulability index $\mu(\boldsymbol{z}, \lambda)$ is evaluated for $N_\lambda \in \mathbb{N}_+$ equi-distant solutions of the null-space. A resulting null-space manipulability grid of size $N_t \times N_\lambda$ is depicted in Figure 4.2.
3. This discrete grid is eventually translated to a graph structure, which forms the basis for the globally optimal redundancy resolution.

While for the first two steps we build on the results of chapters 2 and 3 on the OTG and the efficient manipulability map respectively, the third part applies globally optimal graph search techniques. In the following, first all three steps are discussed in detail and eventually we present a framework to integrate the approach in an overall SE(3) OTG architecture.

Algorithm 4.1: Resolve redundancy along a SE(3) end-effector trajectory

```
Function resolveRedundancy
    Data   : $\boldsymbol{x}^{\text{cur}}$, $\lambda^{\text{cur}}$, $\boldsymbol{x}^{\text{des}}$, $T^{\text{pred}}$, $\Delta t$, $N_\lambda$, $\mathcal{C}$
    Result : $\lambda^{\text{pred}}(t)$
    $\boldsymbol{x}^{\text{pred}}(t) \leftarrow$ predictTrajectory ($\boldsymbol{x}^{\text{cur}}$, $\boldsymbol{x}^{\text{des}}$, $T^{\text{pred}}$, $\Delta t$, $\mathcal{C}$) ;    // cf. Algorithm 4.2
    $\mathcal{M} \leftarrow$ createManipulapilityMap ($\boldsymbol{x}^{\text{pred}}(t)$, $N_\lambda$) ;    // cf. Algorithm 4.3
    $\lambda^{\text{pred}}(t) \leftarrow$ findGlobalOptimum ($\mathcal{M}$, $\lambda^{\text{cur}}$, $N_\lambda$, $\Delta t$, $\mathcal{C}$) ;    // cf. Algorithm 4.4
```

4.3.1 SE(3) Trajectory Generation

A fast OTG is key to enable reactive robot behavior that is capable of adapting to unforeseen events e.g. detected by its sensors. Especially for safety-related aspects, this requires reactions within a single control cycle at typically 1 kHz. While any fast OTG can be used in this first module, we employ our OTG from Chapter 2 that delivers a reactive increment of the end-effector state within 20 µs. To generate full end-effector trajectories, the OTG is applied iteratively with constant step size Δt, until the trajectory reaches the given prediction horizon

T^{pred}. Both of these temporal parameters are user-defined and determine the maximal number of discrete time steps

$$N_t := \frac{T^{\text{pred}}}{\mathit{\Delta} t} + 1. \tag{4.4}$$

A listing of the approach is given in Algorithm 4.2. Note that while the OTG is expected to react within 1 ms, the manipulability map is guaranteed to be smooth along any singularity-free trajectory $\boldsymbol{x}(t)$. This allows for the redundancy resolution to run on a slower sampling rate than the OTG in an online application.

Algorithm 4.2: Predict end-effector trajectory on SE(3) with magnitude constraints

```
Function predictTrajectory
    Data   : $\boldsymbol{x}^{\text{cur}}$, $\boldsymbol{x}^{\text{des}}$, $T^{\text{pred}}$, $\mathit{\Delta} t$, $\mathcal{C}$
    Result : $\boldsymbol{x}^{\text{pred}}(t)$
    $t \leftarrow 0$;
    $\boldsymbol{x}^{\text{pred}}(t) \leftarrow \boldsymbol{x}^{\text{cur}}$;
    repeat
        $t \leftarrow t + \mathit{\Delta} t$;
        $\boldsymbol{x}^{\text{pred}}(t) \leftarrow$ OTG ($\boldsymbol{x}^{\text{pred}}(t)$, $\boldsymbol{x}^{\text{des}}$, $\mathit{\Delta} t$, $\mathcal{C}$) ;      // cf. Chapter 2
    until $t \geq T^{\text{pred}}$;
```

4.3.2 Manipulability Map

Given a full SE(3) state trajectory $\boldsymbol{x}(t) = (\boldsymbol{z}(t), \dot{\boldsymbol{z}}(t))$ of the end-effector, the efficient closed-form task space manipulability formulations from Chapter 3 are applied to evaluate the manipulability measure

$$(\boldsymbol{z}, \lambda) \rightarrow \mu, \quad \text{SE}(3) \times \mathbb{R} \mapsto \mathbb{R}_{+} \tag{4.5}$$

at N_λ equidistant arm angles λ, to form the discrete manipulability map $\mathcal{M}$ of grid size $N_t \times N_\lambda$.

Taking advantage of the analytic FK and IK formulations of the mappings

$$\text{FK} : \boldsymbol{q} \rightarrow (\boldsymbol{z}, \lambda), \quad \mathbb{R}^7 \mapsto \text{SE}(3) \times \mathbb{R} \tag{4.6a}$$

$$\text{IK} : (\boldsymbol{z}, \lambda) \rightarrow \boldsymbol{q}, \quad \text{SE}(3) \times \mathbb{R} \mapsto \mathbb{R}^7 \tag{4.6b}$$

from Section 3.3 additionally enables efficient adjustment of the discrete $\mathcal{M}$ to account for joint limits in the robot configurations.

This is achieved by modifying the manipulability values in $\mathcal{M}$ according to

$$\bar{\mu}(\boldsymbol{z}, \lambda) := (1 - p(\boldsymbol{z}, \lambda))\mu(\boldsymbol{z}, \lambda) \tag{4.7}$$

with the penalty function

$$p(\boldsymbol{z}, \lambda) := \begin{cases} \infty & \text{for } d^{\max}(\boldsymbol{z}, \lambda) \leq 0 \\ e^{-d^{\max}(\boldsymbol{z}, \lambda)\sigma} & \text{else ,} \end{cases} \tag{4.8a}$$

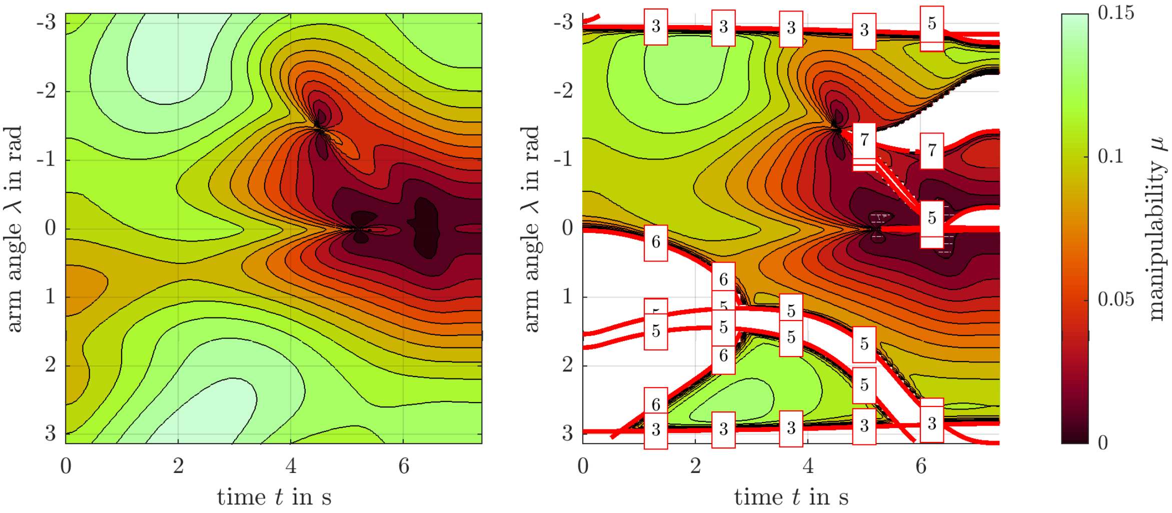


Figure 4.2: Null-space manipulability map before (*left*) and after (*right*) incorporating joint limits and applying penalization (4.8). White areas mark invalid configurations, and the red lines correspond to joint limits w.r.t. the noted joint number.

accounting for the joint limit distance function

$$d^{\max}(\boldsymbol{z}, \lambda) := \min\big(\boldsymbol{q}^{\max} - |\mathrm{IK}(\boldsymbol{z}, \lambda)|\big). \tag{4.8b}$$

A shaping factor of $\sigma = 30$, as used in our implementation, leads to a negligible influence for $d^{\max} > 10°$. This modification invalidates joint limit violating configurations as well penalizes the ones close to such a limit. Figure 4.2 illustrates a penalization of the manipulability map and Algorithm 4.3 lists the essential steps.

Note that the exact analytical definitions of the applied functions μ and IK are fully derived in Section 3.3 and allow for simultaneous vectorized evaluation of the entire grid, which in comparison to iterative evaluation showed a computational performance boost by a factor of 25 and is thus essential for online capable performance.

Algorithm 4.3: Create manipulability map from SE(3) trajectory

```
Function createManipulapilityMap
    Data   : $\boldsymbol{x}^{\text{pred}}(t)$, $N_\lambda$
    Result : $\mathcal{M}$
    $\mathcal{M} \leftarrow \mu(\boldsymbol{x}^{\text{pred}}(t), N_\lambda)$ ;            // cf. Chapter 3
    $\boldsymbol{P} \leftarrow$ calculatePenalty($\mathcal{M}$, $\mathcal{C}$) ;          // cf. (4.8)
    $\mathcal{M} \leftarrow$ applyPenalty($\mathcal{M}$, $\boldsymbol{P}$) ;               // cf. (4.7)
```

4.3.3 Global Optimization

This third module of the approach builds the core part for solving the stated problem of Section 4.2. The fact that we steadily advance in time along the manipulability map naturally results in a directed acyclic graph (DAG) with weighted edges deduced from the given manipulability map. From a practical point of view, the start configuration and thus the end-effector pose and null-space arm angle of the robot are assumed to be known. The exact arm angle at the end of the prediction horizon, on the contrary, is an open degree of freedom for the problem at hand. Treating the manipulability measures at the grid points as distances, interprets the problem at hand as a search for the *shortest path with single source and multiple destinations* in the classical sense of path planning. This allows to apply graph search techniques for solving the redundancy resolution problem, which guarantee a global optimum of the solution. We first outline the translation from a manipulability grid to a graph object, followed by a discussion of applying the actual shortest-path strategy.

4.3.3.1 Create Graph from Map

Given is a discrete grid $\mathcal{M}$, that holds the manipulability measures μ of a set of null-space arm angle parameters λ over a discrete trajectory in time t. In order to apply graph search algorithms, $\mathcal{M}$ is converted to a weighted graph $\mathcal{G} := (V, E)$, defined by the set of vertices V and the set of edges $E := (u, v) \in V^2$ with an assigned positive weight $w(e) : E \mapsto \mathbb{R}$.

The set of points on the discrete $N_t \times N_\lambda$ grid $\mathcal{M}$ is interpreted as the set of vertices V, while the edges $e \in E$ reflect possible trajectory segments of the arm angle. Therefore, the resulting in- and out-degree of the graph, i.e. the maximal number of in- and outgoing edges of each vertex, are

$$\deg^{\text{in}}(\mathcal{G}) := 1 \tag{4.9}$$

$$\deg^{\text{out}}(\mathcal{G}) := 2\mathit{\Delta}\lambda + 1, \tag{4.10}$$

where $\deg^{\text{out}}(\mathcal{G})$ directly reflects the maximal angular velocity of the robot elbow $\omega_\lambda^{\max}$ through

$$\mathit{\Delta}\lambda := \texttt{floor}\left(\omega_\lambda^{\max}\frac{N_\lambda}{2\pi}\mathit{\Delta}t\right) \tag{4.11}$$

and consequently defines the set of edges as

$$E := \{(u, v) | v = u(t+1, \lambda \pm \mathit{\Delta}\lambda), (u, v) \in V^2, \mathit{\Delta}\lambda \in \mathbb{N}_+\}. \tag{4.12}$$

In case of $\mathit{\Delta}\lambda = 0$, the ratio of time and null-space discretizations is not compatible and needs adjustment of the user-parameters in algorithms 4.2 and 4.3, to either slow down the trajectory or allow a higher resolution of the arm angles in $\mathcal{M}$.

With the focus of the presented approach being the computation performance, the according weights of the edges $w(e)$ are chosen to directly correspond to the manipulability at the outgoing vertices u. Because graph search algorithms are traditionally designed to find shortest paths, , we define the weights as however

$$w(e) := \max(\mathcal{M}) - \mathcal{M}(u) + \epsilon \tag{4.13}$$

with $\max(\mathcal{M})$ being the maximal manipulability value in the grid $\mathcal{M}$. An arbitrary offset $\epsilon \in \mathbb{R}_+$ assures $w > 0$, as some graph search algorithms reserve $w = 0$ to indicate invalid edges.

Figure 4.3 illustrates the translation from manipulation map to manipulation graph for a low resolution example.

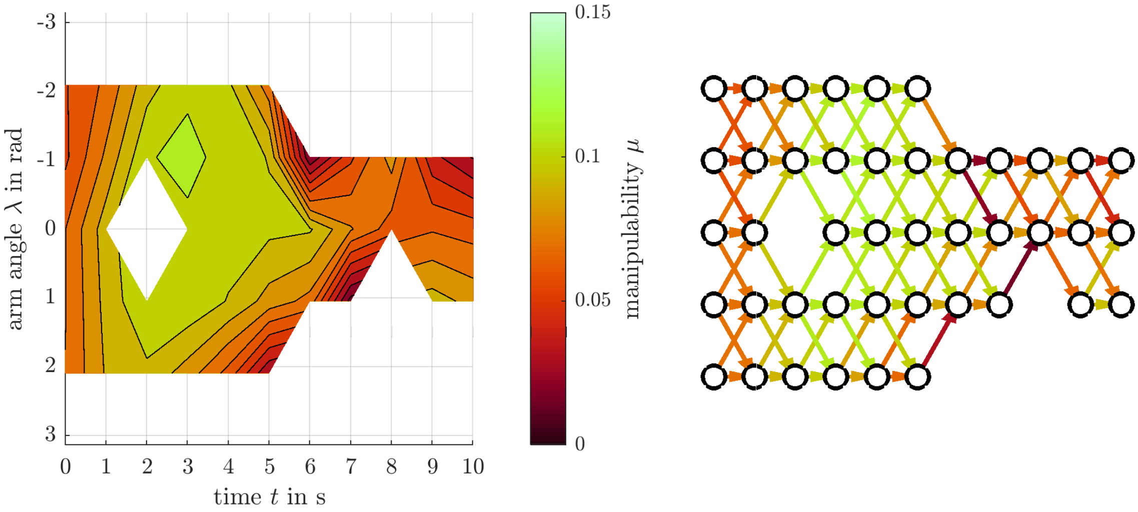


Figure 4.3: Illustration of the graph $\mathcal{G}$ (right) built from the grid $\mathcal{M}$ (left). Note that for demonstration purposes a low temporal as well as spatial resolution with $\Delta t = 1\,\mathrm{s}$ and $N_\lambda = 7$ is chosen. The resulting number of outgoing edges of each vertex is 3 for the case $\omega_\lambda^{\max} = 1$.

4.3.3.2 Find Shortest Path

Once a graph is created, we can draw on a large body of readily available standard graph search algorithms to solve a classical path planning problem.

For offline trajectory generation, the starting as well as end vertices of the path search might be only limited to the corresponding time horizon, resulting in *multiple source and multiple destination* problems. To find the generally best trajectory for both these cases, *all-pair algorithms* such as Floyd-Warshall [Flo62] or Johnson's [Joh77] can be employed. These algorithms deliver shortest paths for every start-end pair, and thus allow to simply select the best solution among the valid candidates. As a special case of these problems, the initially mentioned *cyclic solutions*, are often of interest especially in industrial contexts. Cyclic solutions can be found by directly selecting the best solution that also fulfills $\lambda(t = 0) = \lambda(t = T^{\mathrm{pred}})$.

In online scenarios, on the other hand, the starting vertex of the path search is determined as the one nearest to the current arm angle $\lambda(t = 0)$. The arm angle at the end of the path prediction $\lambda(t = T^{\mathrm{pred}})$ in general stays arbitrary. This results in a *single source and multiple destination* problem with N_λ possible end vertices for the graph search, i.e. all vertices of the last time layer. Prominent algorithms to solve this category of problems are Dijkstra's algorithm [Dij59], with time complexity $\mathcal{O}(|V|\log(|V|) + |E|)$ in modern implementations, or the slower but more general Bellman-Ford algorithm [Bel58] that also allows for negative edge weights at $\mathcal{O}(|V|\,|E|)$, which are not present in our graph setup though. The operator $|\cdot|$ in this notation, refers to the number of elements in a set. Popular extensions of Dijkstra's original algorithm with heuristic-guided search, e.g. A^* [HNR68], do often improve performance of the graph search, see [NKH12] for an algorithm comparison. However, the main heuristic for a *search direction* in 2D-path planning is in our case clearly given – by the physical requirement of forwarding in time – and already encoded in the directed edges E of the graph. Note that Dijkstra's algorithm is strongly inspired by Bellman's *Principle of Optimality*, which is the foundation for Dynamic Programming [Bel52]. Both, conceptually

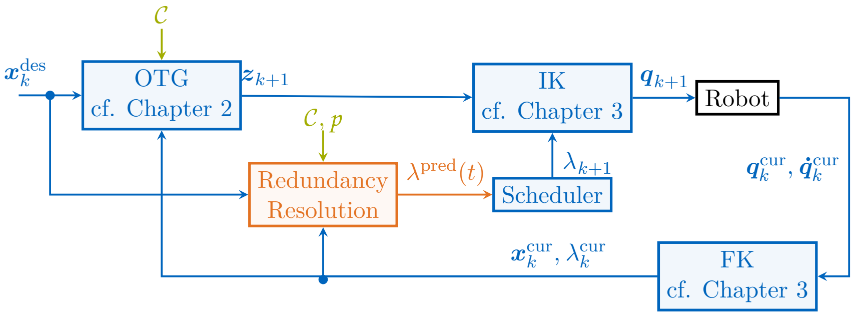

$\mathcal{C}$
$\boldsymbol{x}_k^{\mathrm{des}}$
OTG
cf. Chapter 2
$\boldsymbol{z}_{k+1}$
IK
cf. Chapter 3
$\boldsymbol{q}_{k+1}$
Robot
$\mathcal{C}, p$
Redundancy
Resolution
$\lambda^{\mathrm{pred}}(t)$
λ_{k+1}
Scheduler
$\boldsymbol{q}_k^{\mathrm{cur}}, \dot{\boldsymbol{q}}_k^{\mathrm{cur}}$
$\boldsymbol{x}_k^{\mathrm{cur}}, \lambda_k^{\mathrm{cur}}$
FK
cf. Chapter 3

Figure 4.4: Block diagram of an online trajectory generation architecture incorporating the presented redundancy resolution method. The set $\mathcal{C}$ contains all algorithm-specific constraints and p represents the set of algorithm parameters $\{N_\lambda, \Delta t, T^{\mathrm{pred}}\}$. All blue elements refer to a typical robot control loop sampling time of e.g. 1 kHz, while the presented redundancy resolution runs at slower sampling rate of e.g. 50 Hz, forwarding predicted trajectories to a scheduler.

and technically, it constitutes a dynamic programming successive approximation procedure par excellence, as discussed in [Sni06]. Modern implementations of Dijkstra's algorithm usually allow direct definition of such multiple end points and are thus ideal candidates for our framework. There is, however, another approach that is especially tailored to DAGs, which employs *topological sorting* before searching the graph and can be implemented in linear time complexity $\mathcal{O}(|V| + |E|)$ [CLRS09].

Algorithm 4.4 summarizes the discussed strategy for finding the globally optimal trajectory w.r.t. manipulability.

Algorithm 4.4: Find global optimum for redundancy resolution

```
Function findGlobalOptimum
    Data   : $\mathcal{M}$, $\lambda^{\mathrm{cur}}$, $N_\lambda$, $\Delta t$, $\mathcal{C}$
    Result : $\lambda^{\mathrm{pred}}(t)$
    $\mathcal{G} \leftarrow$ createManipulapilityGraph ($\mathcal{M}$, $N_\lambda$, $\Delta t$, $\mathcal{C}$) ;
    $\lambda^{\mathrm{pred}}(t) \leftarrow$ findShortestPath ($\mathcal{G}$, $\lambda^{\mathrm{cur}}$) ;      // from [Gle20] using 'DAG' algorithm
Function createManipulapilityGraph
    Data   : $\mathcal{M}$, $N_\lambda$, $\Delta t$, $\mathcal{C}$
    Result : $\mathcal{G}$
    $\Delta\lambda \leftarrow$ floor $\left(\omega_\lambda^{\max} \frac{N_\lambda}{2\pi} \Delta t\right)$ ;      // cf. (4.11)
    $\mathcal{G} \leftarrow$ pairVertices($\mathcal{M}$, $\Delta\lambda$) ;      // cf. (4.12)
    $\mathcal{G} \leftarrow$ assignWeights($\mathcal{M}$) ;      // cf. (4.13)
```

4.3.4 Overall OTG Architecture

In this last part of the section, we outline how the presented redundancy resolution strategy is embedded in an overall online trajectory generation architecture. A block diagram is depicted in Figure 4.4, consisting of the OTG directly in SE(3) task space Chapter 2, FK and IK mappings from Chapter 3, and the presented redundancy resolution approach applying the same OTG for its prediction. The redundancy resolution block completes the task space parametrization $\boldsymbol{z}$ with an arm angle λ, to select a specific solution in the null-space of the redundant 7-DOF robot.

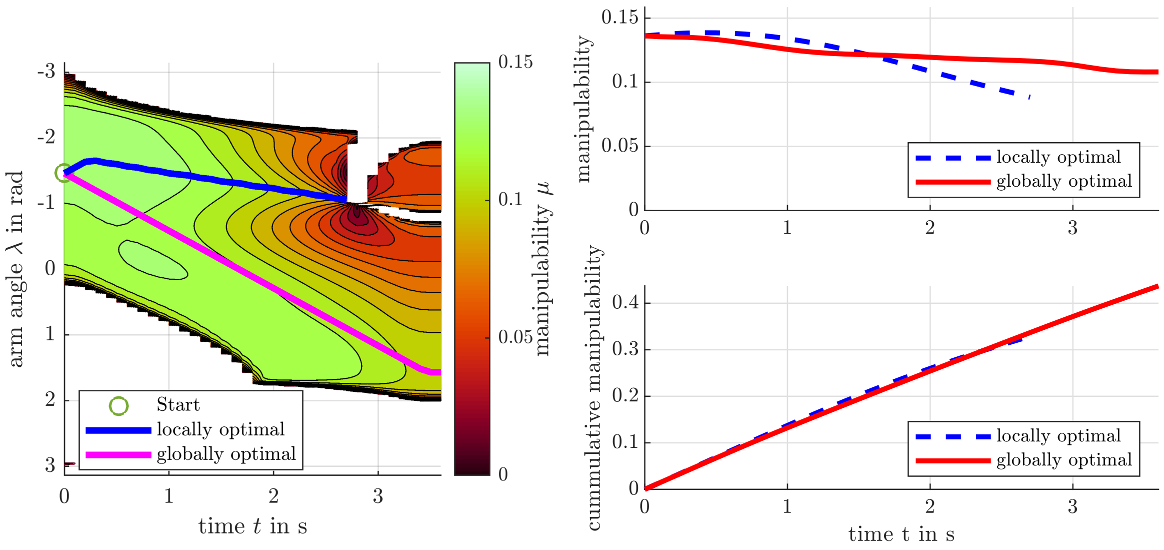


Figure 4.5: Local vs global optimization. The *left* plot shows the manipulability map. While the locally optimizing strategy (blue) follows the optima at $\lambda(t) < 0$ right towards a wrist singularity at $t = 2.8\,\mathrm{s}$, the globally optimal solution (pink) avoids this predicted issue by tolerating sub-optimal configurations in the first 1.5 s. Note that the active $\omega_\lambda^{\max}$ prevents reaching the local optima along the optimum at $\lambda(t) > 0$. The corresponding manipulability metrics of both trajectories are shown on the *right*.

While the OTG ensures reactive behavior of the robot within a single iteration cycle of typically 1 ms, the presented redundancy resolution approach is computationally more demanding and thus runs at a slower sampling frequency of e.g. 20 ms. To handle this discrepancy, a scheduler is used, which receives the predicted discrete trajectory $\lambda^{\mathrm{pred}}(t)$ from the redundancy resolution. It then interpolates the trajectory at the sampling frequency of the robot control loop and rolls out the scheduled trajectory until the new prediction is available.

Note that potential joint velocity limit violations that might occur from the commanding joint trajectory after mapping $(\boldsymbol{z}_{k+1}, \lambda_{k+1}) \mapsto \boldsymbol{q}_{k+1}$, can be handled within the OTG by linear time scaling of the end-effector pose trajectory $\boldsymbol{z}(t)$ as described in Section 2.4.3. Using the IK expressions developed in Section 3.3, the time scaling factor in the OTG can be evaluated analytically, without jeopardizing its hard real-time quality.

4.4 Results

To evaluate the performance of our approach, we first compare it to common local optimization of the manipulability measure, followed by a test for online capability and a time complexity analysis to highlight the influence of the algorithm parameters.

4.4.1 Local vs. Globally Optimal Redundancy Resolution

Unlike conventional local optimizations of the manipulability measure, our approach is capable of transitioning between local optima in the manipulability map, or in critical cases can predict and avoid locally optimal trajectories that inevitably result in singularities. Figure 4.5

illustrates how in such a scenario, global optimization with trajectory prediction outperforms optimization techniques that solely use local information. Start and end poses together with the globally optimal trajectory of this example correspond to the 3D visualization in Figure 4.1, the used parameters are given in Table 4.1.

4.4.2 Time Complexity Analysis

The time complexity of the proposed algorithm parts can be stated in terms of the number of times steps N_t for the trajectory generation algorithm (Algorithm 4.2), and the total number of Vertices and Edges in case of the actual redundancy resolution (Algorithm 4.3 and 4.4). While the number of vertices

$$|V| = N_t N_\lambda \tag{4.14}$$

is given by the grid size of $\mathcal{M}$, the total amount of edges depends on the number of outgoing edges $\deg^{\text{out}}(\mathcal{G})$ per vertex. Considering the definitions (4.9) and (4.11) to compute $\deg^{\text{out}}(\mathcal{G})$, reveals that the number of edges does not depend on the resulting resolution of the time discretization, but solely on its prediction horizon

$$|E| = N_\lambda^2 \omega_\lambda^{\max} T^{\text{pred}}. \tag{4.15}$$

An overview of the time complexity analysis w.r.t. to user-defined algorithm parameters is given in Table 4.1. It shows that the redundancy resolution, apart from the trajectory prediction, is dominated by the number of evaluated arm angles N_λ.

4.4.3 Run-time Example

Besides the efficient iterative trajectory generation in Algorithm 4.2, all steps are available in analytical closed-form expressions, and thus allow for *array operation* in vector-optimized programming languages. Note that array operation is also called *vectorization* in e.g. Matlab. It refers to the exploitation of SIMD instructions of modern CPUs and allows to operate on multiple data points simultaneously. We exploit this technique in the current Matlab implementation for the creation and operation on the manipulability grid $\mathcal{M}$ (Algorithm 4.3), via serialization of all evaluation points into matrix form

$$\begin{bmatrix} \boldsymbol{z}_0 & \cdots & \boldsymbol{z}_{N_t} & \boldsymbol{z}_0 & \cdots & \boldsymbol{z}_{N_t} & \cdots \\ \lambda_1 & \cdots & \lambda_1 & \lambda_2 & \cdots & \lambda_2 & \cdots \end{bmatrix}, \tag{4.16}$$

as well as for the creation of the graph $\mathcal{G}$ in Algorithm 4.4, represented in adjacency matrix form. For finding the shortest path in the graph, we apply the Matlab package *Matlab-BGL* [Gle20], which essentially is a Matlab wrapper for the very efficient C++ Boost Graph Library (BGL). Additionally, the presented algorithms are auto-compiled to C++ code with Matlab Coder™. The numbers listed in Table 4.1 were evaluated in Matlab R2020a on an AMD Ryzen 5 PRO 3500U CPU @2.1 GHz and 14 GB RAM, using the Matlab utility function `timeit`. While a single OTG iteration is calculated in under 20 µs, generating a full prediction as discussed in Section 4.3.1 takes 2 ms. Evaluating $\mathcal{M}$ from Section 4.3.2 for the full trajectory prediction and discarding invalid configurations, takes 7 ms for $|V| = 9000$ evaluation points. See Section 3.4.2 for a detailed run-time analysis of this particular step.

Redundancy Resolution Parameters $\boldsymbol{p}$		
Sampling time	Δt	$0.1\,\mathrm{s}$
Prediction horizon	T^{pred}	$5\,\mathrm{s}$
Null-space discretization	N_λ	180
(Resulting time discretization	$N_t = \frac{T^{\mathrm{pred}}}{\Delta t}$	50)
Redundancy Resolution Constraints $\mathcal{C}$		
Arm angle velocity limit	$\omega_\lambda^{\max}$	$1\,\frac{\mathrm{rad}}{\mathrm{s}}$
Joint angle limits	$q_i^{\max}$	$[170, 120, \cdots, 170]^\top\,\frac{\mathrm{deg}}{\mathrm{s}}$
End-Effector Trajectory Constraints (cf. Chapter 2)		
Translational velocity	$v_{\max}$	$0.5\,\frac{\mathrm{m}}{\mathrm{s}}$
Translational acceleration	$a_{\max}$	$1\,\frac{\mathrm{m}}{\mathrm{s}^2}$
Angular velocity	$\omega_{\max}$	$0.5\,\frac{\mathrm{rad}}{\mathrm{s}}$
Angular acceleration	$\alpha_{\max}$	$1\,\frac{\mathrm{rad}}{\mathrm{s}^2}$

Algorithm Analysis w.r.t. parameters		
Function	Complexity	Runtime
`predictTrajectory`	$\mathcal{O}(N_t)$	$2\,\mathrm{ms}$
`createManipulapilityMap`	$\mathcal{O}(N_t N_\lambda)$	$7\,\mathrm{ms}$
`createManipulapilityGraph`	$\mathcal{O}(N_\lambda^2 \omega_\lambda^{\max} T^{\mathrm{pred}})$	$6\,\mathrm{ms}$
`findShortestPath`	$\mathcal{O}(N_t N_\lambda + N_\lambda^2 \omega_\lambda^{\max} T^{\mathrm{pred}})$	$1\,\mathrm{ms}$

Table 4.1: Run-time analysis of the redundancy resolution algorithm

Finding the globally optimal trajectory along the $|E| = 112\,500$ edges of $\mathcal{G}$ takes about the same amount of computation time for this particular set of parameters. However, the shortest path itself is found in only 1 ms. Due to the simple structure of the graph, the choice of algorithm i.e. Dijkstra or the topological sorting for DAGs, does not show a significant difference in run-time.

4.4.4 Comparison to Offline Approaches

The online capability of our approach allows application in a closed-loop architecture (cf. Figure 4.4). The advantage over existing offline approaches for global trajectory optimization is analog to that of model predictive control (MPC) over open-loop optimal control. In particular it is the capability to adapt to deviations from the prediction due to unmodeled behavior or unforeseen events.

4.5 Chapter Conclusion

A method for globally optimal redundancy resolution of the null-space along an end-effector pose trajectory on SE(3) was presented. This was achieved by exploiting the results of our previous publications on a robot platform agnostic online trajectory generation and the efficient evaluations of a task space manipulability map with all associated kinematics mappings for a serial 7-DOF robot. Although approaches to find globally optimal trajectories in this context are usually computationally very expensive, our approach possesses online capabilities. This allows to escape null-space solutions of merely local manipulability optima as well as avoiding kinematic singularities by searching the entire null-space along a full trajectory prediction. The result is an online capable framework, that generates intuitive end-effector trajectories directly on SE(3) and resolves the redundancy in a globally optimal fashion w.r.t. manipulability, to ensure safety-critical reactive robot behavior.

Part II

Elastic Dynamics

Nomenclature of Part II

This chapter follows the convention of using lowercase bold variables for vectors and uppercase bold variables for matrices. All non-bold variables are scalars. Further, subscript annotations are reserved for index notation of multi-dimensional variables as well as expressing partial derivatives, whereas superscript annotations are part of the variable specification. Also note that we omit explicit listing of function parameters whenever it is clear from the context, to not unnecessarily clutter the notation. A list of the most frequently used variables in this work is given.

Beam Parameters

A	area of cross section surface
E	Young's modulus
G	shear modulus
L	total length
ρ	density
c	constant scaling factor of the PDE $c := \frac{EI}{\rho A}$

Vectors

$\hat{\boldsymbol{e}}_i$	base vector of the inertial coordinate system with $i \in [1, 3]$
$\hat{\boldsymbol{g}}_i$	base vector of the coordinate system along the deformed beam $i \in [1, 3]$
$\boldsymbol{f}$	force vector
$\boldsymbol{\kappa}$	curvature with respect to $\hat{\boldsymbol{g}}_i$
$\boldsymbol{m}$	moment vector
$\boldsymbol{r}$	vector from the origin to a point on the deformed beam

Superscripts

${}_{\parallel}(\cdot)$	parallel component
${}_{\perp}(\cdot)$	perpendicular component
$(\cdot)^{\text{ext}}$	external quantity
$(\cdot)^{\text{inertia}}$	quantity due to inertia
$(\cdot)^{\text{internal}}$	internal quantity

Modelling Variables

φ	curve tangent angle between deformed $\hat{\boldsymbol{g}}_1$ and undeformed beam axis $\hat{\boldsymbol{e}}_1$
l	beam parameter along beam axis with range $[0, L]$
t	time
x	first component of location vector along $\hat{\boldsymbol{e}}_1$

y	second component of location vector along $\hat{\boldsymbol{e}}_2$
ϵ	axial tension parameter $\epsilon := \|(\boldsymbol{r})_l\|_2 - 1$
E^{def}	potential/kinetic/external/total Energy with def $\in \{\text{pot}, \text{kin}, \text{ext}, \text{total}\}$ resp.

Finite Element Description

λ	local length parameter with range $[0, 1]$
ψ_i	global orthogonal basis functions in FEM description
$\boldsymbol{\xi}_i$	Hermite local basis functions in FEM description
N	number of elements
$\mathcal{N}_e$	Node with range $e \in [0, N]$
e	element number with range $[1, N]$

Operators

$(\cdot)_x$	partial differentiation of function w.r.t. beam parameter x, i.e. $\frac{\partial(\cdot)}{\partial x}$
$(\cdot)_{ijkl}$	partial differentiation of function w.r.t. parameters i, j, k, l, i.e. $\frac{\partial^4(\cdot)}{\partial i \partial j \partial k \partial l}$
$\delta(\cdot)$	variation of function, e.g. $\delta\varphi$
$\|(\cdot)\|_2$	Euclidean norm $\sqrt{\sum_i (\cdot)_i^2}$
$(\cdot) \circ (\cdot)$	Hadamard or element-wise product

"Natura non facit saltum or, Nature does not make leaps... If you assume continuity, you can open the well-stocked mathematical toolkit of continuous functions and differential equations, the saws and hammers of engineering and physics for the past two centuries (and the foreseeable future)."

— Benoit B. Mandelbrot, in *'The (Mis)behavior of Markets'* (2004)

5

A Concise and Geometrically Exact Planar Beam Model for Arbitrarily Large Elastic Deformation Dynamics

This chapter was previously published in [HWB21]

The potential of large elastic deformations in control applications, e.g. robotic manipulation, is not yet fully exploited, especially in dynamic contexts. Mainly because essential geometrically exact continuum models that are necessary to express these arbitrarily large deformation dynamics, typically result in a set of nonlinear, coupled, partial differential equations that are unsuited for control applications. Due to this lack of appropriate models, current approaches that try to exploit elastic properties are limited to either small deflection assumptions or quasi-static considerations only. To promote further exploration of this exciting research field of large elastic deflection control, we propose a geometrically exact, but yet concise beam model for a planar, shear- and torsion-free case without elongation. The model is derived by reducing the general geometrically exact 3D Simo-Reissner beam model to this special case, where the assumption of inextensibility allows expressing the couple of planar Cartesian parameters in terms of the curve tangent angle of the beam center line alone. We further elaborate on how the necessary coupling between position-related boundary conditions (i.e. clamped and hinged ends) and the tangent angle parametrization of the beam model can be incorporated in a finite element method formulation, and verify all derived expressions by comparison to analytic initial value solutions and an energy analysis of a dynamic simulation result. The presented beam model opens the possibility of designing online feedback control structures for accessing the full potential, that elasticity in planar beam dynamics has to offer.

5.1 Introduction

Dynamic robotic manipulation of highly deformable objects is still a rarely considered field in literature, due to a lack of appropriate dynamic models. Elastic objects in robotic manipulation are usually either considered only quasi-statically [BM14], or dynamically under the assumption of small deflections which in return results in applications where elasticity is often treated as undesired property that needs to be avoided or compensated [Tav15]. There are some approaches that suggest exploiting elasticity instead, e.g. in terms of safety [BT04, HLF^{+}09], payload estimation [MSB15], or energy storage [TK07, HKAS11]. Some authors exploit elasticity also for quasi-static manipulation [McC00], or even to design new strategies for dynamic manipulation, e.g. [PSSB14] or [PP18]. Nonetheless, we propose that elastic dynamics have even more potential that can be exploited when large deflections are considered.

Models that allow taking advantage of such arbitrarily large deformations, need to fulfill two main criteria. They have to

1. describe the dynamics *geometrically exact*, i.e. independently of the magnitude of deformation
2. be concise and simple enough to allow the design of online feedback control

However, such models are not yet available in literature. In this chapter, we start filling this gap for the case of a planar elastic beam undergoing pure bending, by proposing a single-dimensional dynamic continuum model.

5.1.1 Related Work

Large deflection dynamics are typically treated with one of two approaches – discretized approximations using a FEM, or continuum models expressed as PDEs. The FEM-based descriptions, for which recent literature such as [Dur13] reaches real-time capable control, are limited to quasi-static deformable structures, as the computational cost of FEM descriptions for true dynamic large deformation online feedback control is still out of reach. Whereas the geometrically exact continuum models result in multi-variate and highly coupled nonlinear PDE systems and control approaches are thus limited to oscillation damping [Ito01, HT12]. Unlike existing control methods for linear beam models [KSS06, KSSB06], the literature body on theory of nonlinear PDE systems is still too limited in its applicability for the complex expressions arising in these mechanical system models [PA09]. From a control point of view, however, it is well known that continuum models, unlike discretized approximations, do not face so-called spillover phenomena that can lead to instabilities due to unmodeled high frequency dynamics [MB83]. For these reasons, as a first contribution towards filling the discussed gap in literature, this chapter proposes a geometrically exact model of planar Euler-Bernoulli beam dynamics for arbitrarily large deflections, that admits a surprisingly concise PDE formulation.

Although there are special purpose models for large deflection models, such as bullwhip dynamics [BHT58, MG03], or dynamics of a fly fishing line [Spo86], we are interested in more general beam dynamics, that admit rope and cable dynamics as a special case of very low elastic stiffness. A vast literature body exists on 1D analytic beam theories alone. Therefore,

	Beam Model	PDEs	Lateral Bending	Rotary Inertia	Shear Deformation	Axial Torsion	Axial Tension
geometrically exact	Simo-Reissner	six	spatial	✓	✓	(✓)	✓
	Kirchhoff-Love	four	spatial	✗	✗	(✓)	✓
	Kirchhoff	four	spatial	✗	✗	(✓)	✗
	this work	**one**	**planar**	✗	✗	✗	✗
linearized	Timoshenko	two	planar	✓	✓	✗	✗
	Rayleigh	one	planar	✓	✗	✗	✗
	Euler-Bernoulli	one	planar	✗	✗	✗	✗

Table 5.1: Comparison of this work to the most commonly used beam theories. While the proposed model is limited to planar bending, it offers a geometrically exact description in a single-dimensional equation.

because of the considerable computational advantage of 1D beam theories over 3D-continuum mechanics theories, the latter will not be considered in this concise literature review. Starting with the first mathematical treatment of static elasticity by Galileo already in 1638, Hooke's treatise of linear elasticity followed in 1678. The first precise definition of the elastica (a thin strip of elastic material) problem was carried out by Jakob Bernoulli in 1691 [Ber91] and he published its first solution in 1694 [Ber94]. His nephew Daniel Bernoulli [Ber42] did not himself solve the problem, but he suggested Euler the use of variational analysis, who delivers a closed-form solution of the elastica in 1744 [Eul44]. A more detailed and insightful mathematical historical overview of the elastica can be found in [Lev08]. While Euler's early work already predicts slender beam deformations with astounding precision, many authors built on this work to include further effects to account for more general conditions as well as geometries. It is said that Rayleigh [Ray77] added rotary inertia effects in 1877, and Timoshenko [Tim21] further enhanced the theory to account for shear effects. However, Elishakoff [Eli20] discusses original authors and naming of linear beam theories. He e.g. mentions that Bresse [Bre59] already included rotary inertia effects in 1859 and thus before Rayleigh, though the works were developed independently. Further, a beam theory including shear effects was originally published in [Tim16], an earlier book in Russian language, where Timoshenko mentions to have developed the theory together with P. Ehrenfest. Elishakoff therefore suggests the historically justifiable name *Bresse–Rayleigh–Timoshenko–Ehrenfest beam theory*.

For large deflections, also referred to as *finite strain*, geometrically nonlinear models are necessary. Kirchhoff's work from 1859 [Kir59] is a spatial generalization of the *Euler-Bernoulli beam*, and allows modelling of 3D deformations through bending and torsion. The theory was later extended by Love [Lov92] to further account for axial tensions and is referred to as *Kirchhoff-Love beam theory*. Reissner [Rei72] added further measures to Kirchhoffs theory, accounting for shear deformations in planar curves and later for space curves in [Rei81]. Simo [Sim85] enhanced Reissner's work in terms of approximations to what is nowadays known as *Simo-Reissner beam theory* or *geometrically exact beam theory*. To complete the overview it is also worth mentioning that reduced versions of these two well known theories have been proposed, that neglect torsion modes, e.g. [MPW15] for the Kirchhoff-Love and [RUC14] for the Simo-Reissner case. Meier et al. [MPW19] give a more in-depth overview and analysis of these nonlinear beam theories. Table 5.1 shows how our proposed model fills the current literature gap of a concise model for geometrically exact descriptions.

5.1.2 Contribution

The main contribution of this chapter is twofold. We provide

- the first single-dimensional, geometrically exact, dynamic beam model
- a method for incorporating boundary conditions in a FEM formulation, for cases where the FEM model has only descriptive variables of higher order derivatives than the boundary condition itself

5.1.3 Outline

In Section 5.2 we derive the model via step-by-step reduction, starting from the general Simo-Reissner beam theory, extracting first the Kirchhoff-Love beam theory, followed by special case assumptions – isotropic, torsion-free, inextensible and planar bending. The translation into a FEM description is explained in Section 5.3, including our proposal for incorporating position-level boundary conditions into the tangent angle PDE system. This FEM description is applied for the simulation verification in Section 5.4. The chapter is concluded in the final Section 5.5.

5.2 Modelling

This section outlines the model derivation, starting from a general 3D theory. After reducing the model by gradually introducing assumptions, we couple the Cartesian coordinates to achieve an expression in a single PDE.

5.2.1 Model Reduction

The presented geometrically exact model for a planar Euler-Bernoulli beam, is found via reduction of the general 3D Simo-Reissner beam theory. We gradually introduce further assumptions to simplify the dynamic governing equations of the beam. Please note, that we only define the individual components of the equations once necessary, to keep the derivation clear and easy to follow. They are marked in color, to enhance readability. The resulting model forms a special case of a planar Kirchhoff-Love Beam Theory parametrized solely in the curve tangent angle, in analog to Euler's elastica [Eul44]. A more in-detail analysis and discussion of the general Simo-Reissner and Kirchhoff-Love beam theories, including the special cases of isotropic cross-sections and torsion-free formulations can be found in [MPW19].

General Simo-Reissner Beam Theory The first theory that accounts for very general 3D beam deformations including spatial bending, torsion, axial tension and shear deformation, was published by [Sim85]. The consideration of shear effects make it an adequate theory for thick rod dynamics. It is also been denoted *geometrically exact beam theory*, because the description is consistent at the deformed state regardless of the magnitude of displacements, rotations and strains, cf. [CJ99]. Simo himself also used the term *finite strain beam formulation*.

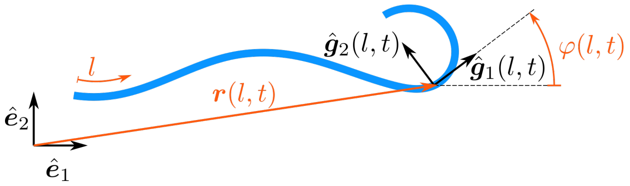


Figure 5.1: Illustration of the used variables to describe beam deformation.

The strong form of the Simo-Reissner beam theory, cf. [Sim85], is a system of 6 coupled PDEs and can be stated with the equilibrium equations

$$\left(\boldsymbol{f}^{\text{internal}}\right)_l + \boldsymbol{f}^{\text{ext}} + \boldsymbol{f}^{\text{inertia}} = \boldsymbol{0} \tag{5.1a}$$

$$\left(\boldsymbol{m}^{\text{internal}}\right)_l + (\boldsymbol{r})_l \times \boldsymbol{f}^{\text{internal}} + \boldsymbol{m}^{\text{ext}} + \boldsymbol{m}^{\text{inertia}} = \boldsymbol{0} \tag{5.1b}$$

where the internal force $\boldsymbol{f}^{\text{internal}}$ and moment vector $\boldsymbol{m}^{\text{internal}}$ result form internal stresses acting on the beam cross-section area at point $\boldsymbol{r}$ of the beam center line. The quantities $\boldsymbol{f}^{\text{ext}}$ and $\boldsymbol{m}^{\text{ext}}$ account for externally imposed forces, and $\boldsymbol{f}^{\text{inertia}}$ and $\boldsymbol{m}^{\text{inertia}}$ are the components due to inertia effects. The detailed constitutive equations that relate $\boldsymbol{f}^{\text{internal}}$ and $\boldsymbol{m}^{\text{internal}}$ to the first Piola-Kirchhoff stress tensor, requires an introduction into 3D continuum mechanics and is omitted in this work. Instead we point the interested reader to related text books such as [Gur82], and define the expressions only after reduction to the specified special cases.

In the following, all objective deformation measures are chosen to be work-conjugated to the material stress resultants in (5.1). We further assume a hyperelastic constitutive relation between these kinetic and kinematic quantities.

Assumption: Vanishing Shear Strains (Kirchhoff-Love Beam Theory) Neglecting shear deformations, and thus assuming that the cross-section is always perpendicular to the center-line of the beam, the change in internal forces $\left(\boldsymbol{f}^{\text{internal}}\right)_l$ can be split up into a parallel component ${}_{\parallel}\left(\boldsymbol{f}^{\text{internal}}\right)_l$ and ${}_{\perp}\left(\boldsymbol{f}^{\text{internal}}\right)_l$, a component perpendicular to the center-line. Further, the moment balance (5.1b) reduces to the projection onto the center-line tangential base vector $\hat{\boldsymbol{g}}_1$, cf. Figure 5.1 for an illustration. The Kirchhoff-Love beam equations are thus given with

$${}_{\parallel}\left(\boldsymbol{f}^{\text{internal}}\right)_l + \underbrace{\left(\frac{(\boldsymbol{r})_l}{\|(\boldsymbol{r})_l\|_2^2} \times \left(\left(\boldsymbol{m}^{\text{internal}}\right)_l + \boldsymbol{m}^{\text{ext}} + \boldsymbol{m}^{\text{inertia}}\right)\right)_l}_{{}_{\perp}(\boldsymbol{f}^{\text{internal}})_l} + \boldsymbol{f}^{\text{ext}} + \boldsymbol{f}^{\text{inertia}} = \boldsymbol{0} \tag{5.2a}$$

$$\hat{\boldsymbol{g}}_1{}^{\top}\left(\left(\boldsymbol{m}^{\text{internal}}\right)_l + \boldsymbol{m}^{\text{ext}} + \boldsymbol{m}^{\text{inertia}}\right) = 0 \tag{5.2b}$$

where (5.2b) is now a scalar expression and the beam model is thus reduced to 4 PDEs.

Assumption: Initially Straight and Isotropic If now an initially straight beam with an isotropic cross-section is assumed for the hyperelastic beam, the components of (5.2) are given with

$$
{}_{\|}\left(\boldsymbol{f}^{\text{internal}}\right)_l + \left(\frac{(\boldsymbol{r})_l}{\|(\boldsymbol{r})_l\|_2^2} \times \left(\underbrace{EI\left(\boldsymbol{\kappa}\right)_l}_{\left(\boldsymbol{m}^{\text{internal}}\right)_l} + \boldsymbol{m}^{\text{ext}} + \boldsymbol{m}^{\text{inertia}} \right) \right)_l + \boldsymbol{f}^{\text{ext}} \underbrace{-\rho A \left(\boldsymbol{r}\right)_{tt}}_{\boldsymbol{f}^{\text{inertia}}} = \boldsymbol{0} \tag{5.3a}
$$

$$
2GI\left(\kappa_1\right)_l + m_1^{\text{ext}} - 2\rho I\left(\omega_1\right)_t = 0, \tag{5.3b}
$$

with the Young modulus E, inertia I, density ρ, cross section area A and the curvature vector

$$
\boldsymbol{\kappa} := \frac{(\boldsymbol{r})_l \times (\boldsymbol{r})_{ll}}{\|(\boldsymbol{r})_l\|_2^2}. \tag{5.4}
$$

The components κ_1, m_1^{ext} and ω_1 in (5.3b) relate to the curvature, externally imposed moment and angular velocity along the tangential direction of the beam.

Assumption: Torsion-Free Assuming pure bending and no torsional effects, the inertia moment and thus the moment balance equation (5.3b) vanishes completely. Moreover only the perpendicular component of the external moment affects the force balance equation

$$
\underbrace{\left(EA\epsilon\hat{\boldsymbol{g}}_1\right)_l}_{{}_{\|}\left(\boldsymbol{f}^{\text{internal}}\right)_l} + \left(\frac{(\boldsymbol{r})_l}{\|(\boldsymbol{r})_l\|_2^2} \times \left(EI\left(\boldsymbol{\kappa}\right)_l + {}_{\perp}\boldsymbol{m}^{\text{ext}} \right) \right)_l + \boldsymbol{f}^{\text{ext}} - \rho A \left(\boldsymbol{r}\right)_{tt} = \boldsymbol{0}, \tag{5.5}
$$

with the axial tension parameter $\epsilon := \|(\boldsymbol{r})_l\|_2 - 1$, which considers that the relation

$$
\left\| \frac{\partial \boldsymbol{r}^{\text{undef}}}{\partial l} \right\|_2 = 1 \tag{5.6}
$$

of the undeformed beam, does not in general hold for the deformed case

$$
\left\| \frac{\partial \boldsymbol{r}}{\partial l} \right\|_2 \neq 1 \tag{5.7}
$$

due to possible elongations in the beam structure.

Assumption: Inextensible Beam If it is assumed that the beam does not undergo axial elongations, the gradient (5.7) does always equal 1. Hence, the axial tension parameter ϵ evaluates to

$$
\epsilon(l) \equiv \left\| \frac{\partial \boldsymbol{r}}{\partial l} \right\|_2 - 1 \equiv 0 \tag{5.8a}
$$

and thus ${}_{\|}\left(\boldsymbol{f}^{\text{internal}}\right)_l$ vanishes. The beam model (5.5) consequently further simplifies to

$$
\left((\boldsymbol{r})_l \times \left(EI\left((\boldsymbol{r})_l \times (\boldsymbol{r})_{ll} \right)_l + {}_{\perp}\boldsymbol{m}^{\text{ext}} \right) \right)_l + \boldsymbol{f}^{\text{ext}} - \rho A \left(\boldsymbol{r}\right)_{tt} = \boldsymbol{0}. \tag{5.8b}
$$

However, unlike the previous assumptions that can be incorporated implicitly with an adequate choice of parametrization variables, it is in general difficult to find such a set of variables that fulfill the inextensibility constraint (5.8a) by construction. A common practice to enforce the equality constraint (5.8a) on the simulation result in a weak sense, i.e. in an integral form instead of point-wise, is by means of extending the weak form of the model equations (5.8b) with a Lagrange multiplier potential, cf. [MPW19].

The following assumption of pure planar bending, however, does again permit a parametrization that fulfills this constraint directly in the strong sense, i.e. for every point along the beam.

Assumption: Pure Planar Bending The last step in the model reduction, is the restriction to pure planar bending. For the remainder of this section, we will switch to a component wise notation in Cartesian coordinates. The beam model from (5.8b) reduces to two coupled PDEs and is fully described by

$$\begin{bmatrix} (y)_l \left(EI \left((x)_l \, (y)_{ll} - (y)_l \, (x)_{ll} \right)_l + {}_\perp m_z^{\text{ext}} \right) \\ -(x)_l \left(EI \left((x)_l \, (y)_{ll} - (y)_l \, (x)_{ll} \right)_l + {}_\perp m^{\text{ext}}{}_z \right) \end{bmatrix}_l + \begin{bmatrix} f_x^{\text{ext}} \\ f_y^{\text{ext}} \end{bmatrix} - \rho A \begin{bmatrix} x \\ y \end{bmatrix}_{tt} = \mathbf{0}, \tag{5.9a}$$

and the additional inextensibility constraint

$$\left\| \begin{bmatrix} (x)_l \\ (y)_l \end{bmatrix} \right\|_2 - 1 \equiv 0 \tag{5.9b}$$

because the third row of the vector equation (5.8b) vanishes for the purely planar problem. Thus the remaining external inputs are forces $\boldsymbol{f}^{\text{ext}}$ in the xy-plane as well as the external moment ${}_\perp m_z^{\text{ext}}$ perpendicular to this plane.

With the curve tangent angle φ, defined as the angle between the tangent vector of the deformed beam center-line $\hat{\boldsymbol{g}}_1$ and its undeformed counterpart $\hat{\boldsymbol{e}}_1$, the beam curvature (5.4) can be expressed as

$$\boldsymbol{\kappa}^{\text{planar}} := \begin{bmatrix} 0 \\ 0 \\ (\varphi)_l \end{bmatrix} \tag{5.10}$$

for the shear-free, planar, inextensible case. Further, it allows to state the geometric identities

$$(x)_l \equiv \cos(\varphi) \tag{5.11a}$$

$$(y)_l \equiv \sin(\varphi) \,. \tag{5.11b}$$

The planar beam model (5.9) can thus be stated as

$$\begin{bmatrix} \sin(\varphi) \left(EI(\varphi)_{ll} + {}_\perp m_z^{\text{ext}} \right) \\ -\cos(\varphi) \left(EI(\varphi)_{ll} + {}_\perp m_z^{\text{ext}} \right) \end{bmatrix}_l + \begin{bmatrix} f_x^{\text{ext}} \\ f_y^{\text{ext}} \end{bmatrix} - \rho A \begin{bmatrix} x \\ y \end{bmatrix}_{tt} = \mathbf{0}, \tag{5.12}$$

a beam model in mixed form containing Cartesian coordinates as well as the curve tangent angle as describing variables. In the remainder of this Section, (5.12) is the starting point to first develop the static case followed by the general dynamic case, both entirely expressed in the curve tangent angle.

5.2.1.1 Static Beam Model Expressed in the Curve Tangent Angle

Only considering solutions in a static equilibrium, the Cartesian acceleration terms in (5.12) vanish and

$$\begin{bmatrix} \sin(\varphi)\left(EI(\varphi)_{ll} + {}_{\perp}m_z^{\text{ext}}\right) \\ -\cos(\varphi)\left(EI(\varphi)_{ll} + {}_{\perp}m_z^{\text{ext}}\right) \end{bmatrix}_l + \begin{bmatrix} f_x^{\text{ext}} \\ f_y^{\text{ext}} \end{bmatrix} = \mathbf{0} \tag{5.13}$$

remains. By computing all derivatives,

$$\begin{bmatrix} \cos(\varphi)(\varphi)_l\left(EI(\varphi)_{ll} + {}_{\perp}m_z^{\text{ext}}\right) + \sin(\varphi)\left(EI(\varphi)_{lll} + \left({}_{\perp}m_z^{\text{ext}}\right)_l\right) \\ \\ \sin(\varphi)(\varphi)_l\left(EI(\varphi)_{ll} + {}_{\perp}m_z^{\text{ext}}\right) - \cos(\varphi)\left(EI(\varphi)_{lll} + \left({}_{\perp}m_z^{\text{ext}}\right)_l\right) \end{bmatrix} + \begin{bmatrix} f_x^{\text{ext}} \\ \\ f_y^{\text{ext}} \end{bmatrix} = \mathbf{0} \tag{5.14}$$

and rotating the equations from their Cartesian xy coordinate system by φ around the z-axis, by pre-multiplying (5.14) with the rotation matrix

$$\boldsymbol{R}_z(\varphi) := \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix}, \tag{5.15}$$

allows extracting the components perpendicular $\perp$ and parallel $\|$ to the beam center line

$$\begin{bmatrix} (\varphi)_l\left(EI(\varphi)_{ll} + {}_{\perp}m_z^{\text{ext}}\right) \\ \\ -(\varphi)_l\left(EI(\varphi)_{lll} + \left({}_{\perp}m_z^{\text{ext}}\right)_l\right) \end{bmatrix} + \begin{bmatrix} {}_{\|}f^{\text{ext}} \\ \\ {}_{\perp}f^{\text{ext}} \end{bmatrix} = \mathbf{0}. \tag{5.16}$$

The fact that no Cartesian xy parameter of the beam description remains but it is rather described in the curve tangent angle *cta* alone, means that the geometric identities (5.11) and hence the planar inextensibility constraint (5.9b) are now implicitly fulfilled by construction. No further treatment such as Lagrangian multipliers are thus necessary, in contrast to the above beam models (5.8b) and (5.9).
In case of an absent external force $\boldsymbol{f}^{\text{ext}}$, the static beam model (5.16) even admits a simple analytic solution. If a non-trivial curvature $(\varphi)_l \neq 0$ is assumed, (5.16) reduces to

$$(\varphi)_{ll} = -\frac{1}{EI}\,{}_{\perp}m_z^{\text{ext}} \tag{5.17}$$

which can be integrated twice and yields a unique solution if boundary conditions are applied.

5.2.1.2 Dynamic Beam Model Expressed in the Curve Tangent Angle

To also fully state the dynamic planar beam model in terms of the curve tangent angle, the Cartesian xy acceleration terms in (5.12) remain to be expressed in terms of the curve tangent angle φ. This is achieved by differentiating the system of equations (5.12) w.r.t. the beam parameter l. Assuming no buckling of the object, x and y have continuous derivatives and thus Schwarz's theorem allows changing the order of the derivations. Applying the geometric identities (5.11) now also to the acceleration terms, leads to

$$\begin{bmatrix} \sin(\varphi)(\phi)_{ll} \\ -\cos(\varphi)(\phi)_{ll} \end{bmatrix}_{ll} + \begin{bmatrix} f_x^{\text{ext}} \\ f_y^{\text{ext}} \end{bmatrix}_l - \rho A \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \end{bmatrix}_{tt} = \mathbf{0} \tag{5.18a}$$

with the auxiliary variable

$$(\phi)_{ll} := EI\,(\varphi)_{ll} + {}_{\perp}m_z^{\mathrm{ext}}, \tag{5.18b}$$

a PDE system entirely expressed in terms of the curve tangent angle φ. As for the static case (5.16), the geometric identities fulfill the inextensibility constraint (5.9b) by construction, thus no special consideration is necessary.
Expanding all partial derivatives and grouping the trigonometric terms, (5.18a) yields

$$\begin{bmatrix} \cos(\varphi)\left(2\,(\varphi)_l\,(\phi)_{lll} + \rho A\,(\varphi)_t^2\right) - \sin(\varphi)\left((\varphi)_l^2\,(\phi)_{ll} - (\phi)_{llll} - \rho A\,(\varphi)_{tt}\right) + \left(f_x^{\mathrm{ext}}\right)_l \\ \sin(\varphi)\left(2\,(\varphi)_l\,(\phi)_{lll} + \rho A\,(\varphi)_t^2\right) + \cos(\varphi)\left((\varphi)_l^2\,(\phi)_{ll} - (\phi)_{llll} - \rho A\,(\varphi)_{tt}\right) + \left(f_y^{\mathrm{ext}}\right)_l \end{bmatrix} = \mathbf{0}. \tag{5.19}$$

Similar to the static case, pre-multiplying the entire system with the rotation matrix $\boldsymbol{R}_z(\varphi)$ from (5.15) again extracts the components parallel and perpendicular to the beam center line. The only acceleration term $(\varphi)_{tt}$, however, appears solely in the perpendicular direction

$$EI\left((\varphi)_l^2\,(\varphi)_{ll} - (\varphi)_{llll}\right) + \left((\varphi)_l^2\;{}_{\perp}m^{\mathrm{ext}}{}_z - \left({}_{\perp}m^{\mathrm{ext}}{}_z\right)_{ll}\right) + \left({}_{\perp}f^{\mathrm{ext}}\right)_l - \rho A\,(\varphi)_{tt} = 0 \tag{5.20}$$

which in the case of no external inputs admits the very concise strong form

$$\boxed{(\varphi)_{tt} = c\left((\varphi)_l^2\,(\varphi)_{ll} - (\varphi)_{llll}\right)} \quad \text{with} \quad c := \frac{EI}{\rho A}, \tag{5.21}$$

as a single PDE governing the beam dynamics in a single parameter φ.

While this reduced model is relevant for PDE controller development, it is not directly applicable for use in simulations. We therefore present in the following section a respective approximation with a system of ordinary differential equations (ODEs), in terms of a FEM formulation.

5.3 FEM Formulation

In this section we outline the development of a FEM simulation procedure, starting from the development of the weak form of the beam model (5.21), without considering external forces. After transforming the integro-differential weak form into a system of nonlinear ODEs of second order in time via a Bubnov-Galerkin approximation, a finite element discretization leads to a simulation procedure.

5.3.1 Weak Form of Large Deformation in Curve Tangent Angle

The weak form of (5.21) is found by multiplying the equation with the test function $\delta\varphi$, and integrating over the beam length $l = [0, L]$

$$\frac{1}{c}\int_0^L (\varphi)_{tt}\,\delta\varphi\,\mathrm{d}l = \int_0^L (\varphi)_l^2\,(\varphi)_{ll}\,\delta\varphi\,\mathrm{d}l - \int_0^L (\varphi)_{llll}\,\delta\varphi\,\mathrm{d}l. \tag{5.22}$$

The use of colors is intended to help following the derivation. A sequence of integrations by parts will lead to the final weak form. In a first intermediate step

$$\int\limits_0^L (\varphi)_l^2\,(\varphi)_{ll}\,\mathrm{d}l = (\varphi)_l^3\Big|_0^L - \int\limits_0^L 2\,(\varphi)_l^2\,(\varphi)_{ll}\,\mathrm{d}l$$

allows to solve the integral

$$\int\limits_0^L (\varphi)_l^2\,(\varphi)_{ll}\,\mathrm{d}l = \frac{1}{3}\,(\varphi)_l^3\Big|_0^L .$$

Using this result, both colored terms in (5.22) result in

$$\int\limits_0^L \left((\varphi)_l^2\,(\varphi)_{ll}\right)\delta\varphi\,\mathrm{d}l = \frac{1}{3}\,(\varphi)_l^3\,\delta\varphi\Big|_0^L - \frac{1}{3}\int\limits_0^L (\varphi)_l^3\,(\delta\varphi)_l\,\mathrm{d}l \tag{5.23a}$$

and

$$\int\limits_0^L (\varphi)_{llll}\,\delta\varphi\,\mathrm{d}l = (\varphi)_{lll}\,\delta\varphi\Big|_0^L - \int\limits_0^L (\varphi)_{lll}\,(\delta\varphi)_l\,\mathrm{d}l \tag{5.23b}$$

$$= (\varphi)_{lll}\,\delta\varphi\Big|_0^L - (\varphi)_{ll}\,(\delta\varphi)_l\Big|_0^L + \int\limits_0^L (\varphi)_{ll}\,(\delta\varphi)_{ll}\,\mathrm{d}l \tag{5.23c}$$

which leads to the final dynamic equations in weak form

$$\frac{1}{c}\int\limits_0^L (\varphi)_{tt}\,\delta\varphi\,\mathrm{d}l = \Big[-\,(\varphi)_{lll}\delta\varphi + (\varphi)_{ll}(\delta\varphi)_l + \frac{1}{3}(\varphi)_l^3\delta\varphi\Big]_0^L - \int\limits_0^L (\varphi)_{ll}\,(\delta\varphi)_{ll} + \frac{1}{3}\,(\varphi)_l^3\,(\delta\varphi)_l\,\mathrm{d}l \tag{5.24}$$

that builds the basis for the following FEM formulation. The function φ as well as the variation $\delta\varphi$ have to be members of the Sobolev space $\mathcal{H}^2$, where

$$\mathcal{H}^k := \left\{ w \mid w \in \mathcal{L}_2, \frac{\partial w}{\partial x} \in \mathcal{L}_2, \dots, \frac{\partial^k w}{\partial x^k} \in \mathcal{L}_2 \right\}$$

with the function space of square integrable functions

$$\mathcal{L}_2 := \left\{ w \mid \int\limits_0^1 w^2\,\mathrm{d}x < \infty \right\},$$

such that they are twice continuously differentiable in l, cf. [Red13]. A common method to choose candidates for φ and its variation $\delta\varphi$ is given by the Bubnov-Galerkin Approximation and eventually leads to a system of ODEs.

5.3.2 Bubnov-Galerkin Approximation

In the sense of the Bubnov-Galerkin method, the function φ as well as the test function $\delta\varphi$ will be approximated by

$$\varphi(l,t) \approx \varphi^h(l,t) = \sum_{i=1}^{n} a_i^\varphi(t)\psi_i(l) \tag{5.25a}$$

$$\delta\varphi(l,t) \approx \delta\varphi^h(l,t) = \sum_{j=1}^{n} b_j^\varphi(t)\psi_j(l) \tag{5.25b}$$

using the same set of n weighted orthogonal spatial basis functions $\psi_{1..n} \in \mathcal{H}^2$, together with n time-dependent scaling coefficients a_i^φ for the approximation of the curve tangent angle, and b_j^φ for the test function. The weak formulation (5.24) thus reads

$$\begin{aligned}\frac{1}{c}\sum_j b_j^\varphi \sum_i (a_i^\varphi)_{tt} \int_0^L \psi_i\psi_j \mathrm{d}l = \sum_j b_j^\varphi \Bigg[-\sum_i a_i^\varphi \left(\int_0^L (\psi_i)_{ll}(\psi_j)_{ll} \mathrm{d}l + (\psi_i)_{lll}\psi_j \Big|_0^L - (\psi_i)_{ll}(\psi_j)_l \Big|_0^L \right) \\ - \frac{1}{3} \int_0^L \left(\sum_i a_i^\varphi (\psi_i)_l \right)^3 (\psi_j)_l \mathrm{d}l - \frac{1}{3} \left(\sum_i a_i^\varphi (\psi_i)_l \right)^3 \psi_j \Bigg|_0^L \Bigg].\end{aligned} \tag{5.26}$$

As the coefficients b_j^φ from the variation (5.25b) are arbitrary, the weak formulation results in the system of n equations

$$\begin{aligned}\sum_i (a_i^\varphi)_{tt} \int_0^L \psi_i\psi_j \, \mathrm{d}l = c \Bigg[-\sum_i a_i^\varphi \left(\int_0^L (\psi_i)_{ll} \, (\psi_j)_{ll} \, \mathrm{d}l + (\psi_i)_{lll} \, \psi_j \Big|_0^L - (\psi_i)_{ll} \, (\psi_j)_l \Big|_0^L \right) \\ - \frac{1}{3} \left(\int_0^L \left(\sum_i a_i^\varphi \, (\psi_i)_l \right)^3 (\psi_j)_l \, \mathrm{d}l - \left(\sum_i a_i^\varphi \, (\psi_i)_l \right)^3 \psi_j \Bigg|_0^L \right) \Bigg]\end{aligned} \tag{5.27}$$

for $j = [1, n]$. Reorganizing the terms and using a matrix representation finally leads to

$$\boldsymbol{M} \left(\boldsymbol{a}^\varphi\right)_{tt} = -c \left(\boldsymbol{F}\boldsymbol{a}^\varphi + \frac{1}{3} \boldsymbol{f}^3(\boldsymbol{a}^\varphi) \right), \tag{5.28a}$$

with the components wise definitions

$$M_{ji} := \int_0^L \psi_i\psi_j \, \mathrm{d}l \tag{5.28b}$$

$$F_{ji} := \int_0^L (\psi_i)_{ll} \, (\psi_j)_{ll} \, \mathrm{d}l + (\psi_i)_{lll} \, \psi_j \Big|_0^L - (\psi_i)_{ll} \, (\psi_j)_l \Big|_0^L \tag{5.28c}$$

$$f_j^3(\boldsymbol{a}^\varphi) := \int_0^L \left(\sum_i a_i^\varphi \, (\psi_i)_l \right)^3 (\psi_j)_l \, \mathrm{d}l - \left(\sum_i a_i^\varphi \, (\psi_i)_l \right)^3 \psi_j \Bigg|_0^L . \tag{5.28d}$$

Note the index order of the matrix component definitions M_{ji} and F_{ji}, that is important for the vector notation in (5.28a). In the context of FEM formulations, a particular choice of piecewise orthogonal basis functions ψ is used.

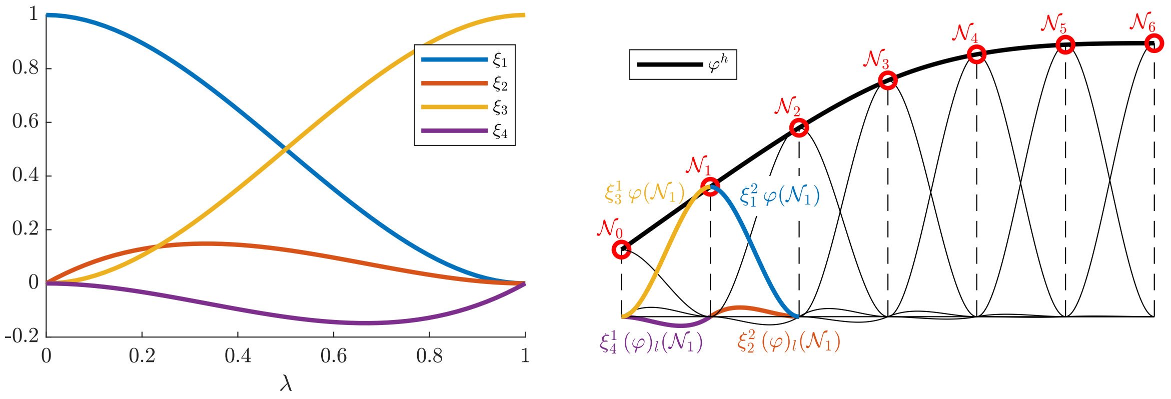


Figure 5.2: FEM nodes and elements. The local basis functions (5.29) of a unit element are shown on the *left*. The *right* plot shows the assembly of $N = 6$ elements to an approximate curve tangent profile φ^h. It further illustrates the role of the local element basis functions $\xi^e_{1..4}$ scaled by function values φ and $(\varphi)_l$ at the node locations $\mathcal{N}_{0..6}$.

5.3.3 Finite Element Discretization

The core idea of the FEM is to discretize the continuous body into a finite number of N elements $e \in [1, N]$, connected at the $N + 1$ node locations $\mathcal{N}_{0..N}$, and choose a special set of piecewise function elements $\psi_{1..N}$ that form the global spline approximation $\varphi^h(l)$ of (5.25a). For the weak formulation (5.24), the orthogonal basis functions need to be at least of class $\mathcal{H}^2$, such that the integral containing second derivatives can be evaluated. In this work we use, for demonstration purposes, the prominent choice of a Hermite cubic splines with cubic elements $\psi_{1..n}$, constructed by the Hermite local cubic polynomial basis functions

$$\begin{bmatrix}\xi_1(\lambda)\\ \xi_2(\lambda)\\ \xi_3(\lambda)\\ \xi_4(\lambda)\end{bmatrix} := \begin{bmatrix}2 & -3 & 0 & 1\\ 1 & -2 & 1 & 0\\ -2 & 3 & 0 & 0\\ 1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}\lambda^3\\ \lambda^2\\ \lambda^1\\ \lambda^0\end{bmatrix}, \qquad \lambda \in [0, 1]. \tag{5.29}$$

They have the special property, that those coefficients that build the spline element

$$\psi_e(\lambda) := \underbrace{\begin{bmatrix}\xi_1^e(\lambda) & \xi_2^e(\lambda) & \xi_3^e(\lambda) & \xi_4^e(\lambda)\end{bmatrix}}_{=:\boldsymbol{\xi}_e^\top(\lambda)}\underbrace{\begin{bmatrix}\varphi(\mathcal{N}_{e-1})\\ (\varphi)_l\,(\mathcal{N}_{e-1})\\ \varphi(\mathcal{N}_e)\\ (\varphi)_l\,(\mathcal{N}_e)\end{bmatrix}}_{=:\boldsymbol{a}_e^\varphi} \tag{5.30}$$

directly correspond to the function value φ and its derivative $(\varphi)_l$ at the node locations $\mathcal{N}_{e-1}$ and $\mathcal{N}_e$. Figure 5.2 illustrates the interplay of the local element basis functions and the function values at the node locations, to approximate the full curve tangent angle function. Although the full dynamics are encoded in the resulting system of ODEs, what is missing for a well-posed problem are the boundary conditions of a particular simulation case.

5.3.4 Boundary Conditions Expressed in the Curve Tangent Angle

Incorporating the boundary conditions on the curve tangent angle φ and its derivatives follows standard procedures, see e.g. [Hug12]. Because treating position-based boundary conditions is not directly possible in the curve tangent angle beam model (5.21), this chapter focuses on the development of a strategy to express boundary conditions in terms of higher order derivatives only. Without loss of generality, we consider for our beam model a fixed end

$$x(l=0,t)=0 \tag{5.31a}$$

$$y(l=0,t)=0 \tag{5.31b}$$

and respectively its dynamic counterpart

$$(x)_{tt}\,(l=0,t)=0 \tag{5.32a}$$

$$(y)_{tt}\,(l=0,t)=0. \tag{5.32b}$$

Fixing an elastic beam in its position at one end, introduces point-wise reaction forces from the mounting onto the beam in x and/or y direction. While these boundary conditions are straight forward to be incorporated in a FEM formulation for a model in the parameters x and y, e.g. (5.9a), the FEM description of the reduced model (5.28) directly acts on the tangent angle function φ and its derivatives, thus not offering any parameter to incorporate position boundary conditions. However, except for the a-priori known position at the boundary condition itself, the curve tangent angle dynamics (5.21) do not require any position parameters to govern the beam profile.

We thus propose to transform the position boundary condition at node $\mathcal{N}_0$, which cannot be incorporated directly, into a dynamic boundary condition for the neighboring node $\mathcal{N}_1$ entirely expressed in curve tangent coefficients $\boldsymbol{a}^{\varphi}$. To define this substitutional boundary condition, we first derive another FEM formulation for the beam model parametrized in Cartesian coordinates (5.9a). Not considering external forces for simplicity, and recalling the geometric identities (5.11), the beam equations (5.9a) reduce to

$$\begin{bmatrix}(x)_{tt}\\(y)_{tt}\end{bmatrix}=c\begin{bmatrix}\sin(\varphi)\,(\varphi)_{ll}\\-\cos(\varphi)\,(\varphi)_{ll}\end{bmatrix}_l. \tag{5.33}$$

Formulating the weak forms and applying another integration by parts reads

$$\frac{1}{c}\int_0^L (x)_{tt}\,\delta x\,\mathrm{d}l=-\int_0^L \sin(\varphi)\,(\varphi)_{ll}\,(\delta x)_l\,\mathrm{d}l+\sin(\varphi)\,(\varphi)_{ll}\,\delta x\Big|_0^L \tag{5.34a}$$

$$\frac{1}{c}\int_0^L (y)_{tt}\,\delta y\,\mathrm{d}l=\int_0^L \cos(\varphi)\,(\varphi)_{ll}\,(\delta y)_l\,\mathrm{d}l-\cos(\varphi)\,(\varphi)_{ll}\,\delta y\Big|_0^L \tag{5.34b}$$

and the Bubnov-Galerkin approximation

$$x(t,l)\approx x(t,l)^h=\sum_{i=1}^{n}a_i^x(t)\psi_i(l)\qquad \delta x(t,l)\approx\delta x(t,l)^h=\sum_{i=1}^{n}b_i^x(t)\delta\psi_i(l) \tag{5.35a}$$

$$y(t,l)\approx y(t,l)^h=\sum_{i=1}^{n}a_i^y(t)\psi_i(l)\qquad \delta y(t,l)\approx\delta y(t,l)^h=\sum_{i=1}^{n}b_i^y(t)\delta\psi_i(l) \tag{5.35b}$$

with the same set of orthogonal functions ψ, leads to the systems of equations

$$\frac{1}{c}\boldsymbol{M}\left(\boldsymbol{a}^x\right)_{tt} = \boldsymbol{f}^x\left(\boldsymbol{a}^x\right) \tag{5.36a}$$

$$\frac{1}{c}\boldsymbol{M}\left(\boldsymbol{a}^y\right)_{tt} = \boldsymbol{f}^y\left(\boldsymbol{a}^y\right) \tag{5.36b}$$

with the components

$$M_{ji} = \int\limits_0^L \psi_j \psi_i \,\mathrm{d}l \tag{5.37a}$$

$$f_j^x = -\int\limits_0^L \sin\left(\varphi\right)\left(\varphi\right)_{ll}\left(\delta\psi_j\right)_l \,\mathrm{d}l \tag{5.37b}$$

$$f_j^y = \int\limits_0^L \cos\left(\varphi\right)\left(\varphi\right)_{ll}\left(\delta\psi_j\right)_l \,\mathrm{d}l. \tag{5.37c}$$

Note that for both systems of equations in (5.36), the matrices $\boldsymbol{M}$ are the same as for the FEM in the curve tangent angle φ from (5.28).

Again using Hermite cubic spline basis functions (5.30), the equations of interest in the two systems of N equations (5.36) are the ones relating to the position basis function of $\mathcal{N}_1$, i.e. $j = 3$:

$$\frac{1}{c}\sum_i \int\limits_{\mathcal{N}_0}^{\mathcal{N}_2} \psi_3 \psi_i \,\mathrm{d}l \,\left(a_i^x\right)_{tt} = -\int\limits_{\mathcal{N}_0}^{\mathcal{N}_2} \sin\left(\varphi\right)\left(\varphi\right)_{ll}\left(\psi_3\right)_l \,\mathrm{d}l \tag{5.38a}$$

$$\frac{1}{c}\sum_i \int\limits_{\mathcal{N}_0}^{\mathcal{N}_2} \psi_3 \psi_i \,\mathrm{d}l \,\left(a_i^y\right)_{tt} = \int\limits_{\mathcal{N}_0}^{\mathcal{N}_2} \cos\left(\varphi\right)\left(\varphi\right)_{ll}\left(\psi_3\right)_l \,\mathrm{d}l \tag{5.38b}$$

As illustrated in Figure 5.2, this requires and integration from $\mathcal{N}_0$ until $\mathcal{N}_2$, to fully account for all associated local basis functions and thus only involves the first six coefficients of the acceleration vectors

$$\left(\boldsymbol{a}_{1..6}^x\right)_{tt} := \begin{bmatrix} (x)_{tt}\left(\mathcal{N}_0\right) \\ (x)_{ltt}\left(\mathcal{N}_0\right) \\ (x)_{tt}\left(\mathcal{N}_1\right) \\ (x)_{ltt}\left(\mathcal{N}_1\right) \\ (x)_{tt}\left(\mathcal{N}_2\right) \\ (x)_{ltt}\left(\mathcal{N}_2\right) \end{bmatrix} \quad \text{and} \quad \left(\boldsymbol{a}_{1..6}^y\right)_{tt} := \begin{bmatrix} (y)_{tt}\left(\mathcal{N}_0\right) \\ (y)_{ltt}\left(\mathcal{N}_0\right) \\ (y)_{tt}\left(\mathcal{N}_1\right) \\ (y)_{ltt}\left(\mathcal{N}_1\right) \\ (y)_{tt}\left(\mathcal{N}_2\right) \\ (y)_{ltt}\left(\mathcal{N}_2\right) \end{bmatrix} \tag{5.39}$$

relating to the function values and first derivatives at the first three nodes. While the right hand side of the FEM formulations (5.36) is already defined in the curve tangent angle φ, what remains is to also rewrite coefficient vectors $\left(\boldsymbol{a}_{1..6}^x\right)_{tt}$ and $\left(\boldsymbol{a}_{1..6}^y\right)_{tt}$ in terms of φ instead of x and y. Starting again from the geometric identities $(x)_l \equiv \cos\left(\varphi\right)$ and $(y)_l \equiv \sin\left(\varphi\right)$, the coefficients in (5.39) can be expressed as

$$(x)_{ltt}(l,t) := \qquad - \sin(\varphi(l,t))\,(\varphi(l,t))_{tt} \qquad - \cos(\varphi(l,t))\,(\varphi(l,t))_t^2 \tag{5.40a}$$

$$(x)_{tt}(l,t) := (x)_{tt}(0,t) - \int_0^l \sin(\varphi(s,t))\,(\varphi(s,t))_{tt}\,\mathrm{d}s - \int_0^l \cos(\varphi(s,t))\,(\varphi(s,t))_t^2\,\mathrm{d}s \tag{5.40b}$$

$$(y)_{ltt}(l,t) := \qquad \cos(\varphi(l,t))\,(\varphi(l,t))_{tt}\,\mathrm{d}s \qquad - \sin(\varphi(l,t))\,(\varphi(l,t))_t^2 \tag{5.40c}$$

$$(y)_{tt}(l,t) := (y)_{tt}(0,t) + \int_0^l \cos(\varphi(s,t))\,(\varphi(s,t))_{tt}\,\mathrm{d}s - \int_0^l \sin(\varphi(s,t))\,(\varphi(s,t))_t^2\,\mathrm{d}s \tag{5.40d}$$

containing Volterra integrals with an upper limit l. Recalling the spline approximation φ^h from (5.30) and using it for the acceleration terms

$$(\varphi)_{tt} = \sum_e (\psi_e)_{tt} = \sum_e \boldsymbol{\xi}_e^\top (\boldsymbol{a}_e^\varphi)_{tt} \tag{5.41}$$

allows to express the entire FEM balance equations (5.38) in terms of φ^h with the additional boundary values $(x)_{tt}(0,t)$ and $(y)_{tt}(0,t)$ which are, however, known a-priori from the position boundary condition we are incorporating. Colors in (5.40a) group the acceleration-related terms, that are used to assemble the left hand side of (5.38), and the components that form the load vector i.e. the corresponding right hand side. Eventually, (5.38) can be evaluated entirely in φ with the left-hand sides

$$\boldsymbol{M}_{3,1..6}\left(\boldsymbol{a}^x_{3,1..6}\right)_{tt} = \underbrace{\boldsymbol{M}_{3,1..6} \circ \begin{bmatrix} 0 \\ -\sin(\varphi(\mathcal{N}_0,t))(\boldsymbol{\xi}_0)_{tt}^\top(\lambda{=}0,t) \\ -\int_0^{\mathcal{N}_1} \sin(\varphi(l,t))(\boldsymbol{\xi}_0)_{tt}^\top(l,t)\mathrm{d}s \\ -\sin(\varphi(\mathcal{N}_1,t))(\boldsymbol{\xi}_1)_{tt}^\top(\lambda{=}0,t) \\ -\int_0^{\mathcal{N}_1} \sin(\varphi(l,t))(\boldsymbol{\xi}_1)_{tt}^\top(l,t)\mathrm{d}l - \int_{\mathcal{N}_1}^{\mathcal{N}_2} \sin(\varphi(l,t))(\boldsymbol{\xi}_2)_{tt}^\top(l-\mathcal{N}_1,t)\mathrm{d}l \\ -\sin(\varphi(\mathcal{N}_2,t))(\boldsymbol{\xi}_2)_{tt}^\top(\lambda{=}0,t) \end{bmatrix}^\top}_{=:\boldsymbol{M}^x_{3,1..6}} (\boldsymbol{a}^\varphi_{1..6})_{tt} \tag{5.42a}$$

and

$$\boldsymbol{M}_{3,1..6}\left(\boldsymbol{a}^y_{3,1..6}\right)_{tt} = \underbrace{\boldsymbol{M}_{3,1..6} \circ \begin{bmatrix} 0 \\ \cos(\varphi(\mathcal{N}_0,t))(\boldsymbol{\xi}_0)_{tt}^\top(\lambda{=}0,t) \\ \int_0^{\mathcal{N}_1} \cos(\varphi(l,t))(\boldsymbol{\xi}_0)_{tt}^\top(l,t)\mathrm{d}l \\ \cos(\varphi(\mathcal{N}_1,t))(\boldsymbol{\xi}_1)_{tt}^\top(\lambda{=}0,t) \\ \int_0^{\mathcal{N}_1} \cos(\varphi(l,t))(\boldsymbol{\xi}_1)_{tt}^\top(l,t)\mathrm{d}l \int_{\mathcal{N}_1}^{\mathcal{N}_2} \cos(\varphi(l,t))(\boldsymbol{\xi}_2)_{tt}^\top(l-\mathcal{N}_1,t)\mathrm{d}l \\ \cos(\varphi(\mathcal{N}_2,t))(\boldsymbol{\xi}_2)_{tt}^\top(\lambda{=}0,t) \end{bmatrix}^\top}_{=:\boldsymbol{M}^y_{3,1..6}} (\boldsymbol{a}^\varphi_{1..6})_{tt} \tag{5.42b}$$

where $\circ$ denotes an element-wise product, considering the piecewise function element definitions $(\psi_e)_{tt}$ from (5.30) and the local Hermite cubic base functions $\boldsymbol{\xi}_e(\lambda, t)$ expressed in the global arc length coordinate $\boldsymbol{\xi}_e(l, t) := \boldsymbol{\xi}_e(\lambda = l/N, t)$. Note that only the first six columns of $\boldsymbol{M}$ affect the position value at $\mathcal{N}_1$, all remaining entries of the third row of $\boldsymbol{M}$ are thus zero. The right hand side load value in x direction of $\mathcal{N}_1$ expressed in φ reads

$$f_3^x(\varphi) := -\int_{\mathcal{N}_0}^{\mathcal{N}_2} \sin(\varphi) + \boldsymbol{M}_{3,1..6} \begin{bmatrix} -(x)_{tt}(0,t) \\ \cos(\varphi(\mathcal{N}_0,t))\,(\varphi)_t\,(\mathcal{N}_0,t)^2 \\ \int_0^{\mathcal{N}_1} \cos(\varphi(s,t))\,(\varphi(s,t))_t^2\;\mathrm{d}s \\ \cos(\varphi(\mathcal{N}_1,t))\,(\varphi)_t\,(\mathcal{N}_1,t)^2 \\ \int_0^{\mathcal{N}_2} \cos(\varphi(s,t))\,(\varphi(s,t))_t^2\;\mathrm{d}s \\ \cos(\varphi(\mathcal{N}_2,t))\,(\varphi)_t\,(\mathcal{N}_2,t)^2 \end{bmatrix} \tag{5.42c}$$

and the load value in y direction reads

$$f_3^y(\varphi) := \int_{\mathcal{N}_0}^{\mathcal{N}_2} \cos(\varphi) + \boldsymbol{M}_{3,1..6} \begin{bmatrix} -(y)_{tt}(0,t) \\ \sin(\varphi(\mathcal{N}_0,t))\,(\varphi)_t\,(\mathcal{N}_0,t)^2 \\ \int_0^{\mathcal{N}_1} \sin(\varphi(s,t))\,(\varphi(s,t))_t^2\;\mathrm{d}s \\ \sin(\varphi(\mathcal{N}_1,t))\,(\varphi)_t\,(\mathcal{N}_1,t)^2 \\ \int_0^{\mathcal{N}_2} \sin(\varphi(s,t))\,(\varphi(s,t))_t^2\;\mathrm{d}s \\ \sin(\varphi(\mathcal{N}_2,t))\,(\varphi)_t\,(\mathcal{N}_2,t)^2 \end{bmatrix}. \tag{5.42d}$$

These expressions are used to incorporate the position boundary conditions into the FEM formulation of the curve tangent angle beam equation.

The resulting system of nonlinear ODEs is used to verify the proposed beam model as well as the presented strategy for incorporating position boundary conditions.

5.4 Simulation Verification

The proposed beam model of Section 5.2 together with the method for incorporating boundary conditions on lower-level derivatives than the descriptive variables of the FEM formulation from Section 5.3.4, are verified in simulation. First, the developed FEM strategy is tested in two different an initial value problems and demonstrated for different magnitudes of external forces and momenta. The second part verifies a dynamic simulation in terms of energy consistency of the beam profile. For both cases, we consider the case of a clamped end at $l = 0$

and a free end at $l = L$. This is expressed in the boundary conditions

$$\begin{aligned} x(l = 0, t) &= 0 && (5.43a)\\ y(l = 0, t) &= 0 && (5.43b)\\ \varphi(l = 0, t) &= 0 && (5.43c)\\ (\varphi)_\lambda\,(l = L, t) &= 0 && (5.43d)\\ (\varphi)_{\lambda\lambda}\,(l = L, t) &= 0. && (5.43e) \end{aligned}$$

While the essential boundary condition (5.43c) and natural boundary conditions (5.43d) and (5.43e) are directly considered in the FEM formulation with conventional techniques, the conditions (5.43a) and (5.43b) are incorporated with the expressions developed in Section 5.3.4. All simulations have been conducted in MATLAB R2020a.

5.4.1 Initial Value Problem

The static solutions of the FEM formulation (5.28a) with external nodal forces $\boldsymbol{f}$ext

$$\mathbf{0} = -c\left(\boldsymbol{F}\boldsymbol{a}^{\varphi} + \frac{1}{3}\boldsymbol{f}^{3}(\boldsymbol{a}^{\varphi}) + \boldsymbol{f}\text{ext}\right), \tag{5.44}$$

are verified in two scenarios. First, the analytic solution of (5.17) is replicated with an external moment at the free end, and in the second example an external nodal force $\boldsymbol{f}^{\text{ext}}$ is applied to the middle of the beam. The initial value problem for both cases is formulated as the nonlinear least-squares problem

$$\min_{\boldsymbol{a}^{\varphi}} \left\| \begin{bmatrix} -c\left(\boldsymbol{F}\boldsymbol{a}^{\varphi} + \frac{1}{3}\boldsymbol{f}^{3}(\boldsymbol{a}^{\varphi}) + \boldsymbol{f}\text{ext}\right) \\ f_3^x \\ f_3^y \end{bmatrix} \right\|_2^2, \tag{5.45}$$

where the last two rows impose the boundary conditions (5.43a) and (5.43b) using the expressions from (5.42c) and (5.42d). The results shown, stem from MATLAB's nonlinear least-squares solver `lsqnonlin()` for a nominal beam of $L = 1\,\text{m}$ and material parameter $c = 1$, with a FEM discretization into 10 beam elements.

5.4.1.1 External Moment

Applying an external nodal moment at the free end of the beam, results in

$$(\varphi)_{ll} = 0 \tag{5.46}$$

and thus a constant curvature $(\varphi)_l$ along the entire beam, according to the analytic solution (5.17). This is also consistent with analytical solutions known in literature [Ant05]. To verify the proposed formulation with this test case, various external momenta proportional to the material parameters ${}_{\perp}m_z^{\text{ext}}(l = L) \in \{0.5, 1, 2, 4\}\pi/EI$ in N m are applied to the free end of the nominal beam. They are incorporated directly as boundary condition $(\varphi)_l\,(l = L) = {}_{\perp}m_z^{\text{ext}}$, substituting (5.43d). The results are shown in Figure 5.3 and depict segments of a perfect circle, due to the constant curvature $(\varphi)_l$. While the $(x)_l$ and $(y)_l$

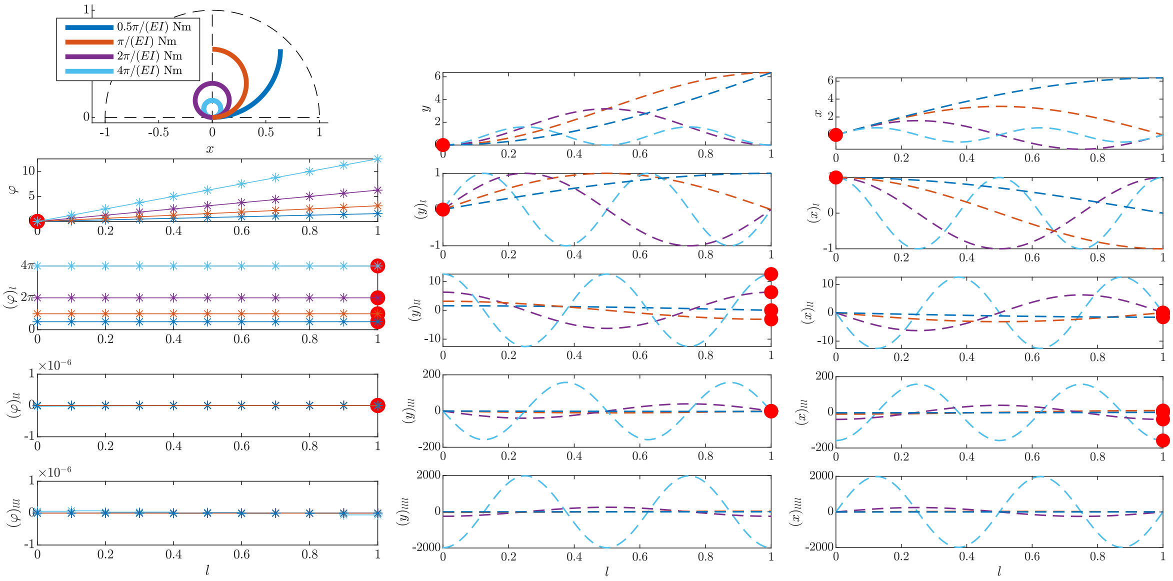


Figure 5.3: Results of the initial value problem, simulated in the beam tangent angle φ for different external momenta at the free end $l = L$. The results of the actual initial value calculation of φ profiles is shown on the left. Additionally, the middle and right columns show the resulting x and y profiles (dashed lines). The large red dots, mark the imposed boundary conditions on the curve tangent angle profile and their respective impact in x and y direction. Note that the x and y profiles, as well as the 2D visualization on the top left, are evaluated during post-processing of the actual simulation results in the curve tangent angle φ, applying the geometric identities (5.11) and respective derivatives.

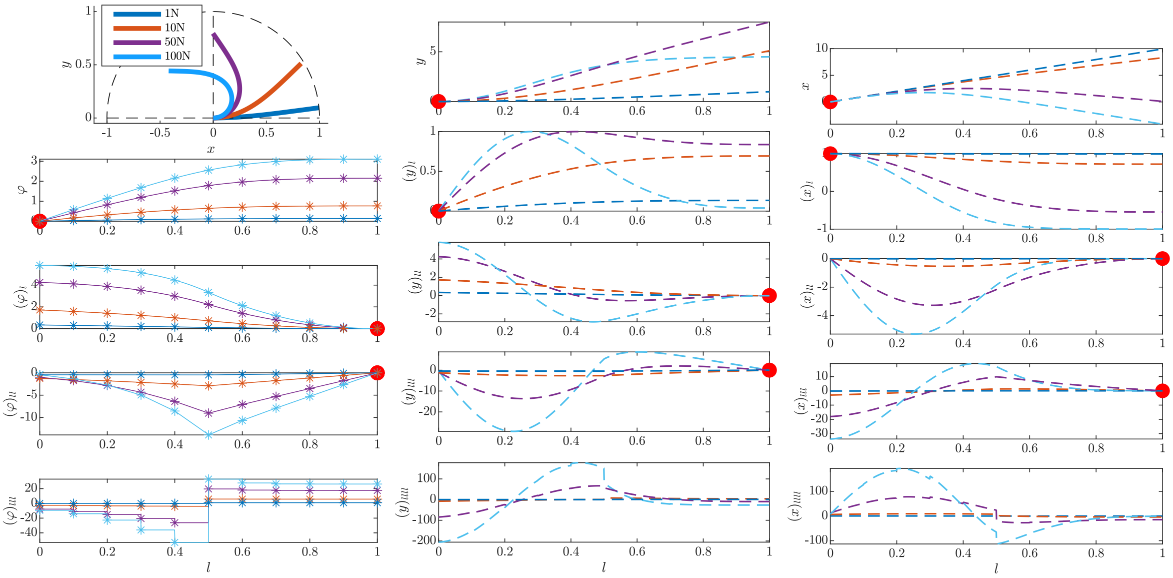


Figure 5.4: Results of the initial value problem, simulated in the beam tangent angle φ for different external nodal forces at $l = 0.5$. The results of the actual initial value calculation of φ profiles is shown on the left. A detailed explanation of the illustration is given in the caption of Figure 5.3.

components show a FEM approximation of the $\cos(\cdot)$ and $\sin(\cdot)$ functions respectively, the according angle φ is a purely linear function in this special case and thus can be approximated with arbitrary accuracy, even for a low number of elements. Note that the analytical trigonometric solution is not visually distinguishable from the simulation result and thus not shown in Figure 5.3.

5.4.1.2 External Force

This second example shows the beam deflection under different external nodal forces applied to the middle node of the nominal beam $f^{\text{ext}}(l = \frac{L}{2}) \in \{1, 10, 50, 100\}\text{N}$. Physically, this expresses an external force perpendicular to the beam center line, cf. (5.20). Results are depicted in Figure 5.4. While for the two smaller deflections the beam profiles in y direction correspond to results of a common linearized Euler-Bernoulli Beam model, the geometrical nonlinearities show full effect for the two larger deflection cases. Note that the initial curvature at the clamped end $l = 0$, which is proportional to $(\varphi)_l$, is the result of the imposed position boundary condition discussed in Section 5.3.4. This verifies the effect of the proposed strategy on the equilibrium configuration.

5.4.2 Dynamic Simulation

For the dynamic case, the FEM formulation (5.28a) is again extended by external nodal forces

$$\boldsymbol{M}\,(\boldsymbol{a}^{\varphi})_{tt} = -c\left(\boldsymbol{F}\boldsymbol{a}^{\varphi} + \frac{1}{3}\boldsymbol{f}^3(\boldsymbol{a}^{\varphi}) + \boldsymbol{f}^{\text{ext}}\right), \tag{5.47}$$

which allows external perturbation of the system. Assuming that the initial beam configuration $\varphi(l, t = 0)$ complies with the boundary conditions (5.43), the expressions for $(x)_{tt}\,(l = 0, t)$ and $(y)_{tt}\,(l = 0, t)$ as developed in Section 5.3.4 are imposed on the beam, using a Lagrangian multipliers technique. The full system reads

$$\underbrace{\begin{bmatrix} \boldsymbol{M} & \left(\boldsymbol{B}^{\text{BC}}\right)^{\top} \\ \boldsymbol{B}^{\text{BC}} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{M}_{\text{Lag}}} \begin{bmatrix} (\boldsymbol{a}^{\varphi})_{tt} \\ \boldsymbol{\lambda} \end{bmatrix} = -c \begin{bmatrix} \boldsymbol{F}\boldsymbol{a}^{\varphi} + \frac{1}{3}\boldsymbol{f}^3(\boldsymbol{a}^{\varphi}) + \boldsymbol{f}^{\text{ext}} \\ \boldsymbol{g}^{\text{BC}} \end{bmatrix}, \tag{5.48}$$

where the position boundary conditions from (5.42) are encoded in

$$\boldsymbol{B}^{\text{BC}} := \begin{bmatrix} \boldsymbol{M}^x_{3,1..6} & \boldsymbol{0}^{\top} \\ \boldsymbol{M}^y_{3,1..6} & \boldsymbol{0}^{\top} \end{bmatrix} \quad \text{and} \quad \boldsymbol{g}^{\text{BC}} := \begin{bmatrix} f^x_3 \\ f^y_3 \end{bmatrix} \tag{5.49}$$

with the first 6 columns of $\boldsymbol{B}^{\text{BC}}$ containing the components from (5.42a) and (5.42b). The matrix $\boldsymbol{M}_{\text{Lag}}$ is invertible, and thus allows to solve for

$$\begin{bmatrix} (\boldsymbol{a}^{\varphi})_{tt} \\ \boldsymbol{\lambda} \end{bmatrix} = -c\boldsymbol{M}_{\text{Lag}}^{-1} \begin{bmatrix} \boldsymbol{F}\boldsymbol{a}^{\varphi} + \frac{1}{3}\boldsymbol{f}^3(\boldsymbol{a}^{\varphi}) + \boldsymbol{f}^{\text{ext}} \\ \boldsymbol{g}^{\text{BC}} \end{bmatrix}. \tag{5.50}$$

The resulting acceleration vector of curve tangent angle coefficients $(\boldsymbol{a}^{\varphi})_{tt}$ is then integrated to simulate the time-varying trajectories. The results of Figure 5.4 are computed using Matlab's `ode45` solver. In a post-processing step, the energy distribution in the beam is evaluated w.r.t. time t.

5.4.2.1 Energy Analysis

The dynamic simulation result is verified, by confirming energy consistency of the simulation result. This is achieved by calculating the total energy

$$E^{\text{total}}(t) = E^{\text{pot}}(t) + E^{\text{kin}}(t) + E^{\text{ext}}(t) \tag{5.51}$$

consisting of the potential energy E^{pot}, the kinetic energy E^{kin} and the externally injected energy E^{ext}. While the potential energy is directly proportional to the curvature of the beam center line

$$E^{\text{pot}} = \frac{1}{2}\int_0^L EI\,(\varphi)_l^2 \,\mathrm{d}l, \tag{5.52}$$

it can be directly evaluated from the simulation results. The kinetic energy

$$E^{\text{kin}} = \frac{1}{2}\int_0^L \rho A\left((x)_t^2 + (y)_t^2\right)\mathrm{d}l \tag{5.53}$$

however, needs to be rewritten in terms of φ first. The geometric identities $(x)_l \equiv \cos(\varphi)$ and $(y)_l \equiv \sin(\varphi)$ together with Schwarz's theorem on changing the order of derivatives, again allow to rewrite the Cartesian components at an arc length l of the beam center line, with the Volterra integrals

$$(x)_t = -\int_0^l \sin\left(\varphi(s,t)\right)\left(\varphi(s,t)\right)_t \,\mathrm{d}s \tag{5.54a}$$

$$(y)_t = \int_0^l \cos\left(\varphi(s,t)\right)\left(\varphi(s,t)\right)_t \,\mathrm{d}s \tag{5.54b}$$

and thus the kinetic energy of the entire beam can be evaluated with

$$E^{\text{kin}} = \frac{1}{2}\int_0^L \rho A\left(\left(\int_0^l \sin\left(\varphi(s,t)\right)\left(\varphi(s,t)\right)_t \,\mathrm{d}s\right)^2 + \left(\int_0^l \cos\left(\varphi(s,t)\right)\left(\varphi(s,t)\right)_t \,\mathrm{d}s\right)^2\right)\mathrm{d}l. \tag{5.55}$$

Note that the rotational component of the kinetic energy are not considered, due to the Euler-Bernoulli assumption of slender beams.

5.4.2.2 Simulation Results

For the dynamic case, an initially unloaded beam with a normalized beam material parameter $c = 1$ and unit length $L = 1\,\text{m}$ is perturbed by an external nodal force

$$f^{\text{ext}}\left(l = \tfrac{L}{2}, t\right) := \begin{cases} f\text{ext}_{\max}\sin\left(\frac{t}{t_{\max}}\pi\right)^2 & \text{for } t < t_{\max} \\ 0 & \text{else} \end{cases} \tag{5.56}$$

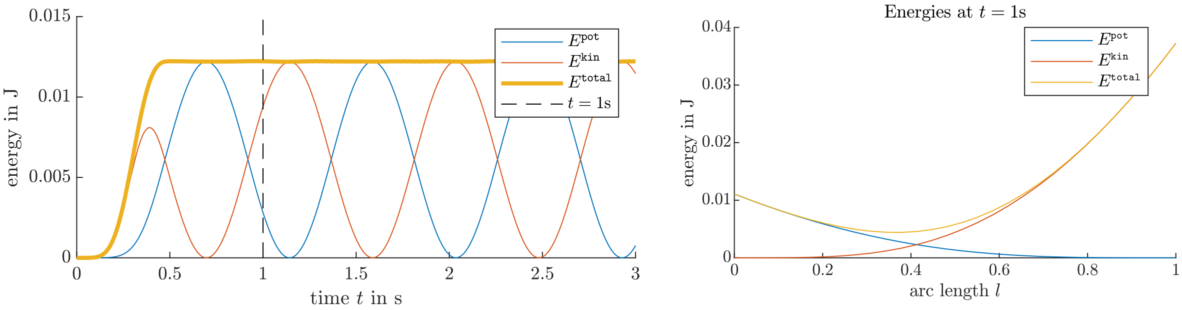


Figure 5.5: Energy conservation during an FEM-simulation of the dynamic curve tangent angle model in a clamped–free scenario, where an external torque applied to the center of an initially straight beam for 0.5 s. The *left* plot shows the energy distribution. The *right* plot shows the energy along the beam at a snapshot taken at t=1 s

again at the beam center, with a maximal unit force $f\text{ext}_{\text{max}} = 1$ over a perturbation time of $t_{\text{max}} = 0.5\,\text{s}$. The result of the dynamic simulation with 10 beam elements is shown in Figure 5.5. While in the first 0.5 s energy is injected in the system via the external perturbation, the resulting beam movement results in an energy exchange between kinetic and potential along the beam, while conserving the total amount of energy in the system. Note that no dissipative terms such as damping are considered in the simulated beam equations (5.21). The right plot of Figure 5.5 shows the energy distribution within the entire beam at $t = 1\,\text{s}$. The accumulation of potential energy at the clamped site of the beam, again demonstrate the effectiveness of the position boundary condition expressions Section 5.3.4.

5.5 Chapter Conclusion

A model for large planar deformation dynamics of Euler-Bernoulli beams was presented and put into context with well known more general beam models. Literature does already offer various models that account for arbitrarily large deformations, however, they typically result in a system of coupled nonlinear PDEs expressions. Whereas the presented approach admits a single-dimensional PDE in one variable, i.e. the curve tangent angle of the beam center line, to describe planar beam dynamics under the common Euler-Bernoulli assumptions of shear-free constant cross-sections.

While boundary conditions on the beam profile derivatives – which is sufficient for sliding and/or free ends – can be directly encoded in a simulation algorithm of the curve tangent angle beam model, there is no descriptive variable available to directly incorporate boundary conditions on the beam position – needed for clamped and/or hinged ends. These cases are, however, of course of highly practical relevance. To also address these cases, we additionally outlined a novel method that allows incorporating boundary conditions in FEM formulations, using solely descriptive variables of higher order derivatives. We apply this method to impose position-based boundary conditions in the FEM formulation of the presented beam model, but it is not limited to solely this use case. The strategy is verified in initial value problems, where it replicates analytical solutions, and a time-variant FEM simulation, by evaluating energy conservation. The presented beam model does not only reduce computational effort due to the dimensional reduction to a single parameter, but the beam profiles are at the same time less complex in this curve tangent angle parametrization and thus require fewer elements in the FEM description.

Although being nonlinear, the derived beam model provides a concise continuum model for future control theory applications of large deformations, where other more complex model descriptions are not appropriate for current model-based PDE controller development. The model reduction process in this chapter, however, also outlines more general beam models considering e.g. shearing, axial torsion, elongation and/or 3D spatial deformations. This chapter is thus also a good source for extended beam models including these additional effects, which might be relevant with the advancement of PDE controller development.

"A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools."
— Douglas Noel Adams, in *'Mostly Harmless'* (1992)

6 Conclusion and Outlook

Dynamic robot manipulation is the supreme discipline in robot manipulation research. It has the capability of achieving additional levels of versatility in a robot manipulation skill set. While this thesis does not contribute to the literature body of robot manipulation skills directly, it addresses two distinct gaps in the literature. The first part comprises algorithms for the generation of agile and reactive robot behavior in human–robot collaboration (HRC) contexts. We addressed central components in this practically relevant problem and proposed an approach to maximize agility of the robot kinematics, while generating end-effector trajectories with proper consideration of human safety aspects w.r.t. the arising HRC standard ISO/TS 15066. Apart from the actual execution of manipulation skills, the second part of this thesis intends to break new ground for the research body on robot manipulation skills, by proposing a new dynamic model of an elastic beam, that can reflect arbitrary large deformation dynamics for the planar case.

6.1 Agile and Reactive Robot End-Effector Motions

The common approach for achieving a versatile set of robot behavior is to develop a library of certain dynamic motion primitives that are then composed and scheduled by a higher-level manipulation planner. One persistent problem, however, is the capability of the robot to execute the necessary joint trajectories. In contrast to purely kinematic or static manipulation skills, movements cannot simply be slowed down as needed, because the manipulation skills typically exploit natural phenomena from environmental contact, friction, gravity, or elasticity. Yet independent of the type of natural dynamics involved in a particular manipulation skill, proposed approaches usually lack applicability in generic contexts. This especially holds true for uncertain dynamic environments such as HRC scenarios where the safety of the human is crucial. In this type of environment it is not predefined when and where in the robot workspace a certain manipulation skill – and consequently a specific end-effector trajectory – is triggered. While manipulation tasks are usually defined as the task space movements of the end-effector, the actual execution of these trajectories with a robotic mechanism eventually requires a mapping into joint space. Present kinematic and dynamic constraints may be further defined in different contexts and need to be addressed simultaneously during online execution. This includes e.g. joint-specific constraints as well as user-defined dynamic constraints that are ultimately dictated by HRC safety standards.

The online trajectory generator (OTG) treatment in Chapter 2 was conducted in a general framework directly acting in the end-effector task space and is therefore to the greatest extend independent of the particular robot kinematics at hand. Only the portion of the developed algorithm that specifically incorporates constraints defined in joint space, considers the actual robot kinematics in use. While there are plenty of approaches available for the translational part of such an OTG, the main contribution stems from the treatment of the complementary general 3D orientation and their mathematical peculiarities. After illustrating the trajectory generation idea for the 3D translational case, we extended the strategy to 3D rotations and derived expressions for the singularity-free unit quaternion as well as rotation matrix representations. Applying the Magnus expansion enables rigorous treatment of trajectory predictions over arbitrary time horizons and results in real-time capable trajectory generation at under 10 µs, hence providing instantaneous reactions to unexpected events.
Furthermore, the nature of our approach naturally allows incorporating constraints directly on the magnitude of physical quantities, thus independent of any coordinate systems. This makes it especially interesting in HRC contexts. We showed how the algorithm can be extended with new constraint definitions in a straight-forward manner, e.g. constraining direction-specific velocities, the total velocity of specific points relative to the tool center point (TCP), total velocity within a safety sphere at the TCP, robot joint velocity constraints, etc. Thus, our approach allows not only for very intuitive definitions of dynamic parameters for the OTG, but directly allows to comply with the emerging HRC safety standards. Note that the strategy is not limited to generate point-to-point (PTP)-movements but is capable of directly forwarding trajectories whenever they comply with the postulated constraints. The approach is therefore directly compatible with higher-level manipulation planners that provide legible paths or manipulation trajectories for the end-effector.

While the proposed OTG per se is robot-agnostic, a practical application in a HRC context eventually introduces specific kinematics to execute the desired end-effector motions. However, the agility of a robot highly depends on the joint configuration of the kinematics and can even result in total loss of maneuverability in the case of kinematic singularities. In Chapter 3 we derived closed-form expressions to evaluate the classical manipulability metric and corresponding joint limits directly from given task space parameters, for the two most common 6-degrees of freedom (DOF) and 7-DOF serial kinematics. This is achieved by introducing a new parameterization of the state- and null-space, that results in concise inverse kinematics (IK) expressions. The closed-form expressions allow for array operation in vector-optimized programming languages, which showed a performance increase of more than an order of magnitude. For large numbers of configurations, i.e. more than 10^5 samples, utilizing graphics processing unit (GPU) computing for massive vectorized evaluations pays off with another order of magnitude increase in performance. An in-depth sensitivity analysis of the manipulability metric w.r.t. task space configurations showed that as serial 7-DOF kinematics has the potential to outperform its 6-DOF counterpart, yet at the same time also admits more joint configurations that result in poor manipulability. The benefit gained from the extra DOF thus relies on a well-conceived utilization of the extra degree of freedom. Further, it was shown that 80 % of the robot poses do not have a unique manipulability maximizing null-space solution, but have up to 4 local optima. This highlights the importance of global over merely local pose optimization. We demonstrated the utility of the developed efficient closed-form expressions in this chapter, by applying it to static global optimization of robot placement w.r.t. a workspace envelope, as well as globally optimal redundancy resolution of a given end-effector pose. The latter shows real-time capabilities at 0.25 ms for an angular resolution of 1°.

The idea of extending the redundancy resolution strategy to entire task space trajectories, as opposed to single independent poses, was further elaborated in Chapter 4. A discrete manipulability grid of null-space solutions over time steps can be evaluated in only a few ms of computation time. It is thus not only useful for offline trajectory planning, but qualifies for online capable redundancy resolution along full task space trajectories, to globally optimize robot joint configurations w.r.t. the manipulability metric. The globally optimal trajectory is found by translating the manipulability grid to a graph structure and applying conventional graph search techniques. In combination with our task space OTG approach from Chapter 2, we proposed a joint trajectory generation framework that does not only provide instant reactions to unforeseen events, but predicts trajectories ahead of time to choose a globally optimal null-space solution that is capable of traversing between local optima and even avoiding potential singularities. Practical parameters for the globally optimal redundancy resolution strategy performs at 20 Hz and can be mixed with the task space OTG to generate IK joint trajectories for typical 1 kHz control loops.

The developed framework can serve as an adaptive low-level joint resolution controller in combination with a potential higher-level manipulation planner in task space. Consequently, it allows the robot to stay maximally agile at all times, not only to preserve the capability of executing manipulation skills, but also to provide safety-critical reaction abilities towards a human co-worker.

6.2 Elastic Dynamics

Various approaches for exploiting natural dynamics from e.g. environmental contact, friction, or gravity were proposed in the literature, but ideas for taking advantage of elasticity are still limited to small deflections or quasi-static cases. One major reason being the complexity of large deflection beam models, which is usually infeasible to handle for the still underdeveloped field of partial differential equation (PDE) control.

We thus proposed a geometrically exact yet concise planar beam model for arbitrarily large elastic deformation dynamics. The model was deduced by gradual reduction of the general Simo-Reissner beam theory, for the special case of a planar, shear- and torsion-free beam without axial elongation. Parametrizing the resulting system of PDEs in the curve tangent angle, instead of Cartesian coordinates, led to a single-dimensional PDE that admits a very concise form. This parametrization, however, comes at the cost of losing the ability to directly incorporate position-related boundary conditions. Yet this severely limits the model to be used in practical use cases. Therefore, the second contribution of this chapter proposes a novel method to address the problem of incorporating boundary conditions in the weak sense, e.g. in finite element method (FEM) formulations, whenever only descriptive variables of higher order derivatives are available.

The proposed PDE model is not restricted to an application in dynamic manipulation, however, it offers a starting point for PDE controller development, due to its concise form and energy-conserving nature without discrete approximations. This lays the foundation for the future development of dynamic manipulation skills that fully exploit internal elastic dynamics.

6.3 Limitations and Future Directions

This last section of the thesis summarizes the current limitations of the presented approaches and points out possible directions for future investigations.

The developed OTG from Chapter 2 inherently constrains velocity and acceleration-related quantities as well as offers an interface to extend the set of constraints that requires consideration. We layed out several ideas tailored for application in an HRC, and also demonstrated how to incorporate joint velocity constraints in the algorithm. Especially for industrial applications, however, an extension to limit jerk in the task space trajectories, or perhaps even a general solution for constraining the trajectory profiles up to an arbitrary order of derivatives, is desirable. Furthermore, the demonstrated strategy for incorporating joint velocity constraints cannot directly be extended to accelerations, due to occurring nonlinearities, and needs further investigation.

The mappings involved in the task space manipulability developed in Chapter 3 result in very well-performing expressions but are all carefully hand-crafted for the specific case of serial 7-DOF robot kinematics. It is not clear how the approach could be generalized to arbitrary – or at least a larger group of – structures. Nonetheless, the discussed 7-DOF kinematic structure, which includes the conventional 6-DOF as a special case, forms a substantial group of robots in use. Besides the elaborated idea of using the task space manipulability expressions for globally optimizing null-space trajectories, we also presented some examples for using the result in static global pose optimization. Due to being of closed-form, the expressions allow to analytically state their derivatives. Future research can take advantage of that, to elaborate on further interesting optimization problems e.g. answering questions regarding robot design, robot placement or end-effector attachment, and more.

The performance of the redundancy resolution approach from Chapter 4 primarily depends on the prediction time horizon, as well as the resolution in time and null-space. Although a practically relevant set of parameters showed already useful performance, there is potential for improvement in terms of explicit incorporation of kinematic constraints. The number of outgoing edges at each point, hence also the size of the graph, is only determined by the maximal angular velocity in the null-space. This, however, does not directly consider possible joint velocity violating trajectories. Although these kinds of violations are indirectly encoded in the manipulability metric, the mentioned closed-form expressions for IK derivatives can be used to prune respective edges in the graph. This directly incorporates joint velocity constraints in the graph and is further expected to improve run-time due to the reduction of the graph size.

The proposed model for elastic beam dynamics in Chapter 5 is a promising candidate for exploration with PDE control theory. Once a control strategy for the presented beam model is elaborated, the task model can be gradually refined by relaxing the introduced assumptions accordingly. Yet there is a lot of work left until a practically relevant use case for a robot manipulation setting is established.

While this thesis contributes two particular aspects towards agile and reactive robot manipulation in HRC, the objective in its entirety continuous to offer an abundance of exciting open research challenges, reaching from perception to low-level control strategies.

“A mathematician is a machine for turning coffee into theorems.”

— Paul Erdős (1913–1996)

A

Remarks on the Online Trajectory Generation Algorithm

A.1 Implementation Remarks

A.1.1 Matrix Logarithm

Although the exponential map, and thus the logarithmic map, are defined as an infinite series

$$\exp(\boldsymbol{R}) := \sum_{n=0}^{\infty} \frac{\boldsymbol{R}^n}{n!} \tag{A.1}$$

in the case of $\boldsymbol{R} \in \mathfrak{so}3$ such as $[\boldsymbol{\omega}]_\times$ there is a closed form solution

$$\exp([\boldsymbol{\omega}]_\times) := I + \sin\left(\|\boldsymbol{\omega}\|_2\right) \frac{[\boldsymbol{\omega}]_\times}{\|\boldsymbol{\omega}\|_2} + \left(1 - \cos\left(\|\boldsymbol{\omega}\|_2\right)\right) \left(\frac{[\boldsymbol{\omega}]_\times}{\|\boldsymbol{\omega}\|_2}\right)^2 \tag{A.2}$$

known as Rodrigues’ rotation formula [MLS94].

A.1.2 Quaternion Logarithm

The following used quaternion functions simplify for the use case in the presented Algorithm. Using quaternions for describing rotations in SO(3), always results in unit quaternions i.e. $\|\mathcal{Q}\|_2 = 1$. If a unit quaternion $\mathcal{Q} \in \mathbb{H}$ is considered to have a scalar and a vector part $\mathcal{Q} = (\mathcal{Q}_w, \mathcal{Q}_v)$ Similar to the closed form matrix expression above, the logarithm for a pure quaternion, i.e. with zero scalar part, admits the concise closed form solution

$$\exp(\mathcal{Q}) := \left(\cos\left(\|\mathcal{Q}_v\|_2\right), \mathcal{Q}_v \sin\left(\frac{\|\mathcal{Q}_v\|_2}{\|\mathcal{Q}_v\|_2}\right)\right) \tag{A.3}$$

and the logarithm of a unit quaternion, i.e. $\|\mathcal{Q}\|_2 = 1$, the logarithm simplifies to

$$\log(\mathcal{Q}) := \left(0, \frac{\phi}{\sin\left(\phi\right) \mathcal{Q}_v}\right) \qquad \text{with } \phi = \arctan 2(\|\mathcal{Q}_v\|_2, \mathcal{Q}_w) \tag{A.4}$$

A.2 Proof that $M(\alpha, t)$ in (2.33) has full rank

Proof. The symbolic calculation of the eigenvalues of $\boldsymbol{M}(\boldsymbol{\alpha}, t)$ leads to

$$\operatorname{eig}(\boldsymbol{M}(\boldsymbol{\alpha}, t)) = \begin{bmatrix} t \\ t - \|\boldsymbol{\alpha}\|_2^2 \frac{t^5}{240} + j \|\boldsymbol{\alpha}\|_2 \frac{t^3}{12} \\ t - \|\boldsymbol{\alpha}\|_2^2 \frac{t^5}{240} - j \|\boldsymbol{\alpha}\|_2 \frac{t^3}{12} \end{bmatrix} \tag{A.5}$$

which is guaranteed to be of full rank, as long as $t \neq 0$. The two contradicting conditions $\boldsymbol{\alpha}$ for rank deficiency are

$$\|\boldsymbol{\alpha}\|_2^2 \frac{t^5}{240} = t \tag{A.6}$$

$$\|\boldsymbol{\alpha}\|_2 \frac{t^3}{12} = 0. \tag{A.7}$$

□

“If God had consulted me when He was creating the world, I would have recommended a simpler system.”
— King Alfonso X of Castile (1221–1284)

B Explicit Task Space Manipulability Mappings

All developed algebraic relations are formulated in a concise enough manner to fully list them in this appendix. Note that $\mathrm{s}(\cdot)$ and $\mathrm{c}(\cdot)$ refer to the $\sin(\cdot)$ and $\cos(\cdot)$ functions resp., to enhance readability.

B.1 Manipulability

The full manipulability mapping M from Section 3.3.1 is given with $\mu(\boldsymbol{q}) = \sqrt{\det\left(\boldsymbol{J}_T \boldsymbol{J}_T{}^\top\right)}$. Note that the Jacobian matrices are formulated w.r.t. the tool frame T at the end-effector. The full symbolic expression for the determinant of the $\mathbb{R}^{6\times 6}$ matrix results in the trigonometric polynomial

$$
\begin{aligned}
\mu(\boldsymbol{q})^2 := 2 r_{SE}{}^2\, r_{EW}{}^2 \left(\mathrm{c}\,(q_4)^2 - 1\right) \Bigg(& \\
&+ r_{SE}{}^2\, \mathrm{c}\,(q_5)^2\, \mathrm{c}\,(q_6)^2 \left(\mathrm{c}\,(q_2)^2 + \mathrm{c}\,(q_4)^2 - \mathrm{c}\,(q_2)^2\, \mathrm{c}\,(q_4)^2 - 1\right) \\
&+ r_{EW}{}^2\, \mathrm{c}\,(q_2)^2\, \mathrm{c}\,(q_3)^2 \left(\mathrm{c}\,(q_4)^2 + \mathrm{c}\,(q_6)^2 - \mathrm{c}\,(q_4)^2\, \mathrm{c}\,(q_6)^2 - 1\right) \\
&+ \left(r_{SE}{}^2 + 2\, r_{SE}\, r_{EW}\, \mathrm{c}\,(q_4)\right) \left(\mathrm{c}\,(q_2)^2 + \mathrm{c}\,(q_6)^2 - \mathrm{c}\,(q_2)^2 \mathrm{c}\,(q_6)^2 - 1\right) \\
&+ \left(r_{SE}{}^2\, \mathrm{s}\,(q_4)\, \mathrm{s}\,(q_6)\, \mathrm{c}\,(q_4)\, \mathrm{c}\,(q_5)\, \mathrm{c}\,(q_6) + r_{SE}\, r_{EW}\, \mathrm{s}\,(q_4)\, \mathrm{s}\,(q_6)\, \mathrm{c}\,(q_5)\, \mathrm{c}\,(q_6)\right) \left(1 - \mathrm{c}\,(q_2)^2\right) \\
&+ \Big(\left(r_{SE}\, r_{EW} + r_{EW}{}^2\, \mathrm{c}\,(q_4)\right) \mathrm{s}\,(q_2)\, \mathrm{s}\,(q_4)\, \mathrm{c}\,(q_2)\, \mathrm{c}\,(q_3) + r_{EW}{}^2\, \mathrm{c}\,(q_2)^2 - r_{EW}{}^2 \Big) \left(1 - \mathrm{c}\,(q_6)^2\right) \\
&\Bigg). \qquad \text{(B.1)}
\end{aligned}
$$

Note that the manipulability measure μ does not depend on joints q_1 nor q_7. Further, the link lengths $\boldsymbol{r}_{BS}$ and $\boldsymbol{r}_{WT}$ do not affect manipulability.

B.2 Inverse Kinematic Functions from (3.43)

$$q_1(\boldsymbol{p},\lambda) := \gamma^{\mathrm{ref}} + \mathrm{atan2}\left(-\,\mathrm{s}\,(\lambda)\;\mathrm{s}\,(\theta_{\mathrm{S}})\,,\;\mathrm{s}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{c}\,(\theta_{\mathrm{S}}) - \mathrm{c}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{c}\,(\lambda)\;\mathrm{s}\,(\theta_{\mathrm{S}})\right) \tag{B.2}$$

$$q_2(\boldsymbol{p},\lambda) := \mathrm{acos}\left(\mathrm{c}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{c}\,(\theta_{\mathrm{S}}) + \mathrm{c}\,(\lambda)\;\mathrm{s}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{s}\,(\theta_{\mathrm{S}})\right) \tag{B.3}$$

$$q_3(\boldsymbol{p},\lambda) := \mathrm{atan2}\left(\mathrm{s}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{s}\,(\lambda)\,,\;\mathrm{c}\,(\lambda)\;\mathrm{s}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{c}\,(\theta_{\mathrm{S}}) - \mathrm{c}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{s}\,(\theta_{\mathrm{S}})\right) \tag{B.4}$$

$$q_4(\boldsymbol{p},\lambda) := \pi - \mathrm{acos}\left(\frac{\frac{{r_{EW}}^2}{2} + \frac{{r_{SE}}^2}{2} - \frac{{r_{\mathrm{ref}}}^2}{2}}{r_{EW}\,r_{SE}}\right) \tag{B.5}$$

$$q_5(\boldsymbol{p},\lambda) := \mathrm{atan2}\left(\mathrm{s}\,(\gamma^{\mathrm{ee}}-\lambda)\;\mathrm{s}\,(\beta^{\mathrm{ee}})\,,\;\mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda) - \mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\right) \tag{B.6}$$

$$q_6(\boldsymbol{p},\lambda) := \mathrm{acos}\left(\mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) + \mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda)\right) \tag{B.7}$$

$$q_7(\boldsymbol{p},\lambda) := \psi^{\mathrm{ee}} + \mathrm{atan2}\left(-\,\mathrm{s}\,(\gamma^{\mathrm{ee}}-\lambda)\;\mathrm{s}\,(\theta_{\mathrm{W}})\,,\;\mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) - \mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda)\right) \tag{B.8}$$

B.3 Absolute Valued Inverse Kinematics Functions from (3.55a)

$$|q_1(\boldsymbol{p},\lambda)| := \gamma_{\mathrm{ref}} + \mathrm{acos}\left(\frac{\mathrm{s}\,(\beta_{\mathrm{ref}})\;\mathrm{c}\,(\theta_{\mathrm{S}}) - \mathrm{c}\,(\beta_{\mathrm{ref}})\;\mathrm{c}\,(\lambda)\;\mathrm{s}\,(\theta_{\mathrm{S}})}{\sqrt{1 - (\mathrm{c}\,(\beta_{\mathrm{ref}})\;\mathrm{c}\,(\theta_{\mathrm{S}}) + \mathrm{c}\,(\lambda)\;\mathrm{s}\,(\beta_{\mathrm{ref}})\;\mathrm{s}\,(\theta_{\mathrm{S}}))^2}}\right) \tag{B.9}$$

$$|q_2(\boldsymbol{p},\lambda)| := \mathrm{acos}\left(\mathrm{c}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{c}\,(\theta_{\mathrm{S}}) + \mathrm{c}\,(\lambda)\;\mathrm{s}\left(\beta^{\mathrm{ref}}\right)\;\mathrm{s}\,(\theta_{\mathrm{S}})\right) \tag{B.10}$$

$$|q_3(\boldsymbol{p},\lambda)| := \pi - \mathrm{acos}\left(\frac{\mathrm{c}\,(\beta_{\mathrm{ref}})\;\mathrm{s}\,(\theta_{\mathrm{S}}) - \mathrm{c}\,(\lambda)\;\mathrm{s}\,(\beta_{\mathrm{ref}})\;\mathrm{c}\,(\theta_{\mathrm{S}})}{\sqrt{1 - (\mathrm{c}\,(\beta_{\mathrm{ref}})\;\mathrm{c}\,(\theta_{\mathrm{S}}) + \mathrm{c}\,(\lambda)\;\mathrm{s}\,(\beta_{\mathrm{ref}})\;\mathrm{s}\,(\theta_{\mathrm{S}}))^2}}\right) \tag{B.11}$$

$$|q_4(\boldsymbol{p},\lambda)| := \pi - \mathrm{acos}\left(\frac{\frac{{r_{EW}}^2}{2} + \frac{{r_{SE}}^2}{2} - \frac{{r_{\mathrm{ref}}}^2}{2}}{r_{EW}\,r_{SE}}\right) \tag{B.12}$$

$$|q_5(\boldsymbol{p},\lambda)| := \pi - \mathrm{acos}\left(\frac{\mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}}) - \mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda)}{\sqrt{1 - (\mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) + \mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda))^2}}\right) \tag{B.13}$$

$$|q_6(\boldsymbol{p},\lambda)| := \mathrm{acos}\left(\mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) + \mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\lambda)\right) \tag{B.14}$$

$$|q_7(\boldsymbol{p},\lambda)| := \psi^{\mathrm{ee}} + \mathrm{acos}\left(\frac{\mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) - \mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\mathrm{la})}{\sqrt{1 - (\mathrm{c}\,(\beta^{\mathrm{ee}})\;\mathrm{c}\,(\theta_{\mathrm{W}}) + \mathrm{s}\,(\beta^{\mathrm{ee}})\;\mathrm{s}\,(\theta_{\mathrm{W}})\;\mathrm{c}\,(\gamma^{\mathrm{ee}}-\mathrm{la}))^2}}\right) \tag{B.15}$$

B.4 Admissible Null-Space Parameter Functions from (3.56)

$$\boldsymbol{\lambda}_1^{\max}(\theta_{\mathrm{S}}, \gamma^{\mathrm{ref}}, \beta^{\mathrm{ref}}, q^{\max}) :=$$
$$\left\{ \begin{aligned} \pm\left(\pi - \operatorname{acos}\left(\frac{\sqrt{\mathrm{s}(\theta_{\mathrm{S}})^2 - \mathrm{s}(\gamma_{\mathrm{ref}} - q^{\max})^2\,\mathrm{s}(\beta_{\mathrm{ref}})^2} + \mathrm{c}(\beta_{\mathrm{ref}})\ \mathrm{s}(\beta_{\mathrm{ref}})\ \mathrm{c}(\theta_{\mathrm{S}})\ \mathrm{s}(\theta_{\mathrm{S}})\left(1 - \mathrm{c}(\gamma_{\mathrm{ref}} - q^{\max})^2\right)}{|\mathrm{s}(\theta_{\mathrm{S}})|\,|\mathrm{c}(\gamma_{\mathrm{ref}} - q^{\max})|\,\mathrm{s}(\theta_{\mathrm{S}})^2\left(\mathrm{s}(\gamma_{\mathrm{ref}} - q^{\max})^2\,\mathrm{s}(\beta_{\mathrm{ref}})^2 - 1\right)}\right)\right) \\ \pm\left(\operatorname{acos}\left(\frac{\sqrt{\mathrm{s}(\theta_{\mathrm{S}})^2 - \mathrm{s}(\gamma_{\mathrm{ref}} - q^{\max})^2\,\mathrm{s}(\beta_{\mathrm{ref}})^2} + \mathrm{c}(\beta_{\mathrm{ref}})\ \mathrm{s}(\beta_{\mathrm{ref}})\ \mathrm{c}(\theta_{\mathrm{S}})\ \mathrm{s}(\theta_{\mathrm{S}})\left(\mathrm{c}(\gamma_{\mathrm{ref}} - q^{\max})^2 - 1\right)}{|\mathrm{s}(\theta_{\mathrm{S}})|\,|\mathrm{c}(\gamma_{\mathrm{ref}} - q^{\max})|\,\mathrm{s}(\theta_{\mathrm{S}})^2\left(\mathrm{s}(\gamma_{\mathrm{ref}} - q^{\max})^2\,\mathrm{s}(\beta_{\mathrm{ref}})^2 - 1\right)}\right)\right) \end{aligned} \right\} \tag{B.16}$$

$$\boldsymbol{\lambda}_2^{\max}(\theta_{\mathrm{S}}, \beta^{\mathrm{ref}}, q^{\max}) := \left\{\ \pm \operatorname{acos}\left(\frac{\mathrm{c}(q^{\max}) - \mathrm{c}(\beta_{\mathrm{ref}})\ \mathrm{c}(\theta_{\mathrm{S}})}{\mathrm{s}(\beta_{\mathrm{ref}})\ \mathrm{s}(\theta_{\mathrm{S}})}\right)\ \right\} \tag{B.17}$$

$$\boldsymbol{\lambda}_3^{\max}(\theta_{\mathrm{S}}, \beta^{\mathrm{ref}}, q^{\max}) :=$$
$$\left\{ \begin{aligned} \pm\left(\pi - \operatorname{acos}\left(\frac{\sqrt{-\mathrm{c}(\beta_{\mathrm{ref}})^2 - \mathrm{c}(q^{\max})^2\,\mathrm{c}(\theta_{\mathrm{S}})^2 + \mathrm{c}(q^{\max})^2 + \mathrm{c}(\theta_{\mathrm{S}})^2} + \mathrm{c}(\beta_{\mathrm{ref}})\ \mathrm{c}(\theta_{\mathrm{S}})\ \mathrm{s}(\theta_{\mathrm{S}})\left(1 - \mathrm{c}(q^{\max})^2\right)}{|\mathrm{c}(q^{\max})|\ \mathrm{s}(\beta_{\mathrm{ref}})\left(\mathrm{s}(q^{\max})^2\,\mathrm{s}(\theta_{\mathrm{S}})^2 - 1\right)}\right)\right) \\ \pm\left(\operatorname{acos}\left(\frac{\sqrt{-\mathrm{c}(\beta_{\mathrm{ref}})^2 - \mathrm{c}(q^{\max})^2\,\mathrm{c}(\theta_{\mathrm{S}})^2 + \mathrm{c}(q^{\max})^2 + \mathrm{c}(\theta_{\mathrm{S}})^2} + \mathrm{c}(\beta_{\mathrm{ref}})\ \mathrm{c}(\theta_{\mathrm{S}})\ \mathrm{s}(\theta_{\mathrm{S}})\left(\mathrm{c}(q^{\max})^2 - 1\right)}{|\mathrm{c}(q^{\max})|\ \mathrm{s}(\beta_{\mathrm{ref}})\left(\mathrm{s}(q^{\max})^2\,\mathrm{s}(\theta_{\mathrm{S}})^2 - 1\right)}\right)\right) \end{aligned} \right\} \tag{B.18}$$

$$\boldsymbol{\lambda}_5^{\max}(\theta_{\mathrm{W}}, \beta^{\mathrm{ee}}, q^{\max}) := \gamma^{\mathrm{ee}} - \boldsymbol{\lambda}_3^{\max}(\theta_{\mathrm{W}}, \beta^{\mathrm{ee}}, q^{\max}) \tag{B.19}$$
$$\boldsymbol{\lambda}_6^{\max}(\theta_{\mathrm{W}}, \beta^{\mathrm{ee}}, q^{\max}) := \gamma^{\mathrm{ee}} - \boldsymbol{\lambda}_2^{\max}(\theta_{\mathrm{W}}, \beta^{\mathrm{ee}}, q^{\max}) \tag{B.20}$$
$$\boldsymbol{\lambda}_7^{\max}(\theta_{\mathrm{W}}, \gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, q^{\max}) := \gamma^{\mathrm{ee}} - \boldsymbol{\lambda}_1^{\max}(\theta_{\mathrm{W}}, \gamma^{\mathrm{ee}}, \beta^{\mathrm{ee}}, q^{\max}). \tag{B.21}$$

References

[AM16] Alexander R Ansari and Todd D Murphey. Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems. *IEEE Transactions on Robotics*, 32(5):1196–1214, 2016.

[AN20] Mohsen Asgari and Amin Nikoobin. Analysis of optimal dynamic manipulation for robotic manipulator based on pontryagin's minimum principle. *Arabian Journal for Science and Engineering*, 45(11):9159–9169, 2020.

[Ant05] Stuart S Antman. Planar steady-state problems for elastic rods. In *Nonlinear Problems of Elasticity*, chapter 4, pages 95–134. Springer Science & Business Media, New York, USA, 2 edition, 2005.

[Ant09] Gianluca Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5):985–994, 2009.

[AORJ15] M Mahdi Ghazaei Ardakani, Björn Olofsson, Anders Robertsson, and Rolf Johansson. Real-time trajectory generation using model predictive control. In *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.

[AYY04] Karim Abdel-Malek, Wei Yu, and Jingzhou Yang. Placement of robot manipulators to maximize dexterity. *International Journal of Robotics and Automation*, 19(1):6–14, 2004.

[AZI$^+$18] Arash Ajoudani, Andrea Maria Zanchettin, Serena Ivaldi, Alin Albu-Schäffer, Kazuhiro Kosuge, and Oussama Khatib. Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975, 2018.

[BBG16] Pierre Besset, Richard Béarée, and Olivier Gibaru. FIR filter-based online jerk-controlled trajectory generation. In *IEEE International Conference on Industrial Technology (ICIT)*, 2016.

[BCOR09] Sergio Blanes, Fernando Casas, JA Oteo, and José Ros. The magnus expansion and some of its applications. *Physics reports*, 470(5-6):151–238, 2009.

[Bel52] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.

[Bel58] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[Ber91] Jakob Bernoulli. Quadratura curvae, e cujus evolutione describitur inflexae laminae curvatura. *Die Werke von Jakob Bernoulli*, 5:223–227, 1691.

[Ber94] Jakob Bernoulli. Curvatura laminae elasticae. *Acta Eruditorum Lipsiae*, 1694:262–276, 1694.

[Ber42] Daniel Bernoulli. The 26th letter to euler. *Correspondence Mathématique et Physique*, 2:499, October 1742. URL: `http://eulerarchive.maa.org/correspondence/fuss`.

[BG14] Corrado Guarino Lo Bianco and Fabio Ghilardelli. A discrete-time filter for the generation of signals with asymmetric and variable bounds on velocity, acceleration, and jerk. *IEEE Transactions on Industrial Electronics*, 61(8):4115–4125, 2014.

[BHT58] Barry Bernstein, Donald A Hall, and Horace M Trent. On the dynamics of a bull whip. *The Journal of the Acoustical Society of America*, 30(12):1112–1115, 1958.

[Bia17] Corrado Guarino Lo Bianco. An efficient algorithm for the real-time generation of synchronous reference signals. *IEEE Transactions on Industrial Electronics*, 64(6):4621–4630, 2017.

[Bjo18] Emil Bjoerlykhaug. A closed loop inverse kinematics solver intended for offline calculation optimized with ga. *Robotics*, 7(1):7, 2018.

[BK19] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446), 2019.

[BM12] Luigi Biagiotti and Claudio Melchiorri. FIR filters for online trajectory planning with time-and frequency-domain specifications. *IFAC Control Engineering Practice*, 20(12):1385–1399, 2012.

[BM14] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research*, 33(1):48–68, 2014.

[BMM16] Luigi Biagiotti, Claudio Melchiorri, and Lorenzo Moriello. Optimal trajectories for vibration reduction based on exponential filters. *IEEE Transactions on Control Systems Technology*, 24(2):609–622, 2016.

[Boy17] Michael Boyle. The integration of angular velocity. *Advances in Applied Clifford Algebras*, 27(3):2345–2374, 2017.

[Bre59] Jacques Antoine Charles Bresse. *Cours de mecanique appliquee: Re'sistance des mate'riaux et stabilite'des constructions*, volume 1. Mallet-Bachelier, 1859.

[BSHA08] Xavier Broquere, Daniel Sidobre, and Ignacio Herrera-Aguilar. Soft motion trajectory planner for service manipulator robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[BT04] Antonio Bicchi and Giovanni Tonietti. Fast and "soft-arm" tactics [robot arm design]. *IEEE Robotics & Automation Magazine*, 11(2):22–33, 2004.

[Bus04] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[BWB08] Andrea Bauer, Dirk Wollherr, and Martin Buss. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008.

[CCSS92] Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Influence of gravity on the manipulability ellipsoid for robot arms. *Journal of Dynamic Systems, Measurement, and Control*, 114(4):723–727, 1992.

[CD95] 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.

[CFK16] Wan Kyun Chung, Li-Chen Fu, and Torsten Kröger. Trajectory generation and planning. In B. Siciliano and O. Khatib, editors, *Springer handbook of robotics*, chapter 8.9. Springer, 2nd edition, 2016.

[Chi87] Stephen Chiu. Control of redundant manipulators for task compatibility. In *International Conference on Robotics and Automation (ICRA)*, volume 4, pages 1718–1724. IEEE, 1987.

[Chi90] Pasquale Chiacchio. Exploiting redundancy in minimum-time path following robot control. In *American Control Conference (ACC)*, pages 2313–2318. IEEE, 1990.

[Chi97] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[Chi00] Pasquale Chiacchio. A new dynamic manipulability ellipsoid for redundant manipulators. *Robotica*, 18(4):381–387, 2000.

[CJ99] Michael A Crisfield and Gordan Jelenić. Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 455(1983):1125–1147, 1999.

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[CT14] Adria Colomé and Carme Torras. Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements. *IEEE/ASME Transactions on Mechatronics*, 20(2):944–955, 2014.

[CWC92] Byoung Wook Choi, Jong Hwa Won, and Myung Jin Chung. Optimal redundancy resolution of a kinematically redundant manipulator for a cyclic task. *Journal of robotic systems*, 9(4):481–503, 1992.

[DHO10] Behzad Dariush, Ghassan Bin Hammam, and David Orin. Constrained resolved acceleration control for humanoids. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 710–717. IEEE/RSJ, 2010.

[Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[Dix78] Laurence Charles Ward Dixon. The global optimization problem. an introduction. *Toward Global Optimization*, 2:1–15, 1978.

[DJ54] George Charles Devol Jr. Programmed article transfer, 1954.

[DK19] Serkan Dereli and Raşit Köker. A meta-heuristic proposal for inverse kinematics solution of 7-dof serial robotic manipulator: quantum behaved particle swarm algorithm. *Artificial Intelligence Review*, pages 1–16, 2019.

[DP09] Alexandre Dolgui and Anatol Pashkevich. Manipulator motion planning for high-speed robotic laser cutting. *International Journal of Production Research*, 47(20):5691–5715, 2009.

[Dra15] Anca D Dragan. *Legible robot motion planning.* PhD thesis, Carnegie Mellon University, 2015.

[DRP+14] Nikhil Chavan Dafle, Alex Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T , Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Regrasping objects using extrinsic dexterity. In *International Conference on Robotics and Automation (ICRA)*, pages 2560–2560. IEEE, 2014.

[DS17] Kévin Dufour and Wael Suleiman. On integrating manipulability index into inverse kinematics solver. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6967–6972. IEEE/RSJ, 2017.

[Dur13] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *International Conference on Robotics and Automation (ICRA)*, pages 3982–3987. IEEE, 2013.

[DVS01] Aaron D'Souza, Sethu Vijayakumar, and Stefan Schaal. Learning inverse kinematics. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 298–303. IEEE, 2001.

[DWLW17] Khoi Hoang Dinh, Philipp Weiler, Marion Leibold, and Dirk Wollherr. Fast and close to optimal trajectory generation for articulated robots in reaching motions. In *International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE/ASME, 2017.

[DXCR93] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM (JACM)*, 40(5):1048–1066, 1993.

[Eli20] Isaac Elishakoff. Who developed the so-called timoshenko beam theory? *Mathematics and Mechanics of Solids*, 25(1):97–116, 2020.

[ETS14] Ben Ezair, Tamir Tassa, and Zvi Shiller. Planning high order trajectories with general initial and final conditions and asymmetric bounds. *The International Journal of Robotics Research*, 33(6):898–916, 2014.

[Eul44] Leonhard Euler. *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrici latissimo sensu accepti*, volume 1. Marcus-Michael Bousquet, 1744.

[FBVT16] Marco Faroni, Manuel Beschi, Antonio Visioli, and Lorenzo Molinari Tosatti. A global approach to manipulability optimisation for a dual-arm manipulator. In *International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016.

[FC20] Enrico Ferrentino and Pasquale Chiacchio. On the optimal resolution of inverse kinematics for redundant manipulators using a topological analysis. *Journal of Mechanisms and Robotics*, 12(3), 2020.

[FFE$^+$18] Carlos Faria, Flora Ferreira, Wolfram Erlhagen, Sérgio Monteiro, and Estela Bicho. Position-based kinematics for 7-dof serial manipulators with global configuration control, joint limit and singularity avoidance. *Mechanism and Machine Theory*, 121:317–334, 2018.

[Flo62] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962. `doi:10.1145/367766.368168`.

[Fub07] Guido Fubini. Sugli integrali multipli. *Rend. Acc. Naz. Lincei*, 16:608–614, 1907.

[GALH10] A. Guigue, M. Ahmadi, R. Langlois, and M. J. Hayes. Pareto optimality and multiobjective trajectory planning for a 7-dof redundant manipulator. *IEEE Transactions on Robotics*, 26(6):1094–1099, 2010.

[GB10] Oscar Gerelli and Corrado Guarino Lo Bianco. A discrete-time filter for the online generation of trajectories with bounded velocity, acceleration, and jerk. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010.

[GBF85] Andrew Goldenberg, Beno Benhabib, and Robert Fenton. A complete generalized solution to the inverse kinematics of robots. *Journal on Robotics and Automation*, 1(1):14–20, 1985.

[GKNK06] Luis Guilamo, James Kuffner, Koichi Nishiwaki, and Satoshi Kagami. Manipulability optimization for trajectory generation. In *International Conference on Robotics and Automation (ICRA)*, pages 2017–2022. IEEE, 2006.

[Gle20] David Gleich. MatlabBGL, 2020. Retrieved June 9, 2020. URL: `https://www.mathworks.com/matlabcentral/fileexchange/10922-matlabbgl`.

[GLS17] Fei Gao, Yi Lin, and Shaojie Shen. Gradient-based online safe trajectory generation for quadrotor flight in complex environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[GPC17] Jiuchun Gao, Anatol Pashkevich, and Stéphane Caro. Optimization of the robot and positioner motion in a redundant fiber placement workcell. *Mechanism and Machine Theory*, 114:170–189, 2017.

[GT04] Karl Gotlih and Inge Troch. Base invariance of the manipulability index. *Robotica*, 22(4):455–462, 2004.

[Gur82] Morton E Gurtin. *An introduction to continuum mechanics.* Academic press, 1982.

[Had15] Sami Haddadin. Physical safety in robotics. In *Formal modeling and verification of cyber-physical systems*, pages 249–271. Springer, 2015.

[HE18] Kris Hauser and Scott Emmons. Global redundancy resolution via continuous pseudoinversion of the forward kinematic map. *IEEE Transactions on Automation Science and Engineering*, 15(3):932–944, 2018.

[HGW17] Gerold Huber, Volker Gabler, and Dirk Wollherr. An online trajectory generator on SE(3) with magnitude constraints. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2017.

[HKAS11] Sami Haddadin, Kai Krieger, and Alin Albu-Schäffer. Exploiting elastic energy storage for cyclic manipulation: Modeling, stability, and observations for dribbling. In *Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 690–697. IEEE, 2011. `doi:10.1109/CDC.2011.6161022`.

[HLF$^+$09] Sami Haddadin, Tim Laue, Udo Frese, Sebastian Wolf, Alin Albu-Schäffer, and Gerd Hirzinger. Kick it with elasticity: Safety and performance in human–robot soccer. *Robotics and Autonomous Systems*, 57(8):761–775, 2009.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[HR20] François Robert Hogan and Alberto Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. In *Algorithmic Foundations of Robotics XII*, pages 800–815. Springer, 2020.

[HS87] JOHNM Hollerbach and Ki Suh. Redundancy resolution of manipulators through torque optimization. *Journal on Robotics and Automation*, 3(4):308–316, 1987.

[HT12] Gareth Hegarty and Stephen Taylor. Classical solutions of nonlinear beam equations: Existence and stabilization. *SIAM Journal on Control and Optimization*, 50(2):703–719, 2012.

[Hug12] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[HW16] Gerold Huber and Dirk Wollherr. Exploiting elastic dynamics in trajectory optimization. In *International Conference on Robotics and Automation (ICRA), Workshop on Exploiting contact and dynamics in manipulation*. IEEE, 2016.

[HW19] Gerold Huber and Dirk Wollherr. Efficient closed-form task space manipulability for a 7-DOF serial robot. *Robotics*, 8(4):98, 2019. `doi:10.3390/robotics8040098`.

[HW20] Gerold Huber and Dirk Wollherr. An online trajectory generator on SE(3) for human–robot collaboration. *Robotica*, 38(10):1756–1777, 2020. `doi:10.1017/S0263574719001619`.

[HW21] Gerold Huber and Dirk Wollherr. Globally optimal online redundancy resolution for serial 7-dof kinematics along SE(3) trajectories. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[HWB21] Gerold Huber, Dirk Wollherr, and Martin Buss. A concise and geometrically exact planar beam model for arbitrarily large elastic deformation dynamics. *Frontiers in Robotics and AI*, 7:226, 2021. `doi:10.3389/frobt.2020.609478`.

[HWR08] R. Haschke, E. Weitnauer, and H. Ritter. On-line planning of time-optimal, jerk-limited trajectories. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[ISO16] ISO/TS 15066. Robots and robotic devices-collaborative robots. Standard, International Organization for Standardization, Geneva, CH, February 2016.

[Ito01] Kazuho Ito. Numerical study for the stability of a geometrically nonlinear elastic beam with velocity feedback. *Nonlinear Analysis: Theory, Methods & Applications*, 47(6):3813–3821, 2001.

[JLK08] Bong-Huan Jun, Pan-Mook Lee, and Seungmin Kim. Manipulability analysis of underwater robotic arms on rov and application to task-oriented joint configuration. *Journal of Mechanical Science and Technology*, 22(5):887–894, 2008.

[JLLL17] Long Jin, Shuai Li, Hung Manh La, and Xin Luo. Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Transactions on Industrial Electronics*, 64(6):4710–4720, 2017.

[Joh77] Donald B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1):1–13, January 1977. `doi:10.1145/321992.321993`.

[JRCC18] Noémie Jaquier, Leonel Dario Rozo, Darwin G Caldwell, and Sylvain Calinon. Geometry-aware tracking of manipulability ellipsoids. In *Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. `doi:10.15607/RSS.2018.XIV.027`.

[KBW$^+$10] Behnam Kamrani, Viktor Berbyuk, Daniel Wäppling, Xiaolong Feng, and Hans Andersson. Optimal usage of robot manipulators. In *Robot Manipulators Trends and Development*. IntechOpen, 2010.

[KDLS92] Kenneth Kreutz-Delgado, Mark Long, and Homayoun Seraji. Kinematic analysis of 7-dof manipulatorsiter. *The International Journal of Robotics Research*, 11(5):469–481, 1992.

[Kha87] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Journal on Robotics and Automation*, 3(1):43–53, 1987.

[Kir59] Gustav Kirchhoff. Ueber das gleichgewicht und die bewegung eines unendlich dünnen elastischen stabes. *Journal für die reine und angewandte Mathematik*, 1859(56):285–313, 1859.

[KJES16] Ivo Kuhlemann, Philipp Jauer, Floris Ernst, and Achim Schweikard. Robots with seven degrees of freedom: Is the additional dof worth it? In *International Conference on Control, Automation and Robotics (ICCAR)*, pages 80–84. IEEE, 2016.

[KKAK13] Robert Katzschmann, Torsten Kröger, Tamim Asfour, and Oussama Khatib. Towards online trajectory generation considering robot dynamics and torque limits. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[KMP18] Okan Koç, Guilherme Maeda, and Jan Peters. Online optimal trajectory generation for robot table tennis. *Robotics and Autonomous Systems*, 105:121–137, 2018.

[KÖÇE04] Raşit Köker, Cemil Öz, Tarık Çakar, and Hüseyin Ekiz. A study of neural network based inverse kinematics solution for a three-joint robot. *Robotics and Autonomous Systems*, 49(3-4):227–234, 2004.

[Kod93] Daniel E Koditschek. Dynamically dexterous robots. *Robot control: Dynamics, motion planning and analysis*, pages 487–490, 1993.

[Kök13] Raşit Köker. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. *Information Sciences*, 222:528–543, 2013.

[Krö10] Torsten Kröger. On-line trajectory generation in robotic systems. In Bruno Siciliano, Oussama Khatib, and Frans Groen, editors, *Springer Tracts in Advanced Robotics*, pages 38–40. Springer-Verlag Berlin Heidelberg, 2010. `doi:10.1007/978-3-642-05175-3`.

[Krö11] Torsten Kröger. Online trajectory generation: Straight-line trajectories. *IEEE Transactions on Robotics*, 27(5):1010–1016, Oct 2011.

[KSS06] Miroslav Krstic, Antranik A Siranosian, and Andrey Smyshlyaev. Backstepping boundary controllers and observers for the slender timoshenko beam: Part I-design. In *American Control Conference (ACC)*, pages 2412–2417. IEEE, 2006.

[KSSB06] Miroslav Krstic, Antranik A Siranosian, Andrey Smyshlyaev, and Matt Bement. Backstepping boundary controllers and observers for the slender timoshenko beam: Part II—stability and simulations. In *Conference on Decision and Control (CDC)*, pages 3938–3943. IEEE, 2006.

[Kuf04] James J Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *International Conference on Robotics and Automation (ICRA)*, pages 3993–3998. IEEE, 2004.

[KW88] Kazem Kazerounian and Zhaoyu Wang. Global versus local optimization in redundancy resolution of robotic manipulators. *The International Journal of Robotics Research*, 7(5):3–12, 1988.

[KW10] Tortsen Kröger and Friedrich M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 26(1):94–111, Feb 2010.

[LAS16a] Friedrich Lange and Alin Albu-Schäffer. Iterative path-accurate trajectory generation for fast sensor-based motion of robot arms. *Advanced Robotics*, 30(21):1380–1394, 2016.

[LAS16b] Friedrich Lange and Alin Albu-Schäffer. Path-accurate online trajectory generation for jerk-limited industrial robots. *IEEE Robotics and Automation Letters*, 1(1):82–89, Jan 2016.

[LB91] Sukhan Lee and Antal K. Bejczy. Redundant arm kinematic control based on parameterization. In *International Conference on Robotics and Automation (IRCA)*, pages 458–465. IEEE Computer Society, 1991.

[Lee89] Sukhan Lee. Dual redundant arm configuration optimization with task-oriented dual arm manipulability. *IEEE Transactions on Robotics and Automation*, 5(1):78–97, 1989.

[Lev08] Raph Levien. The elastica: a mathematical history. *University of California, Berkeley, Technical Report No. UCB/EECS-2008-103*, 2008.

[LLM08] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Time optimal attitude control for a rigid body. In *IEEE American Control Conference (ACC)*, 2008.

[Llo98] John E Lloyd. Trajectory generation implemented as a non-linear filter. *Technical Report, Department of Computer Science, University of British Columbia, Vancouver, B.C.,Canda*, 1998.

[LM96] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[LM99] 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.

[Lov92] Augustus Edward Hough Love. *A Treatise on the Mathematical Theory of Elasticity.* The University Press, Cambridge, 1892.

[Mag54] Wilhelm Magnus. On the exponential solution of differential equations for a linear operator. *Communications on pure and applied mathematics*, 7(4):649–673, 1954.

[Mas99] Matthew T Mason. Progress in nonprehensile manipulation. *The International Journal of Robotics Research*, 18(11):1129–1141, 1999.

[Mas01] Matthew T Mason. *Mechanics of robotic manipulation.* MIT press, 2001.

[MB83] L Meirovitch and Haim Baruh. On the problem of observation spillover in self-adjoint distributed-parameter systems. *Journal of Optimization Theory and Applications*, 39(2):269–291, 1983.

[McC00] Brenan J. McCarragher. A hybrid position/force approach to the exploitation of elasticity in manipulation. In Dominik Henrich and Heinz Wörn, editors, *Robot Manipulation of Deformable Objects*, pages 91–109. Springer London, London, 2000. `doi:10.1007/978-1-4471-0749-1_7`.

[MG03] Tyler McMillen and Alain Goriely. Whip waves. *Physica D: Nonlinear Phenomena*, 184(1):192–225, 2003.

[ML93] Matthew T Mason and Kevin M Lynch. Dynamic manipulation. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 152–159. IEEE, 1993.

[MLS94] Richard M Murray, Z. Li, and S. Sastry. *A mathematical introduction to robotic manipulation.* CRC press, 1994.

[MMAC16] Raymond H Myers, Douglas C Montgomery, and Christine M Anderson-Cook. *Response surface methodology: process and product optimization using designed experiments.* John Wiley & Sons, 2016.

[MMZ$^+$19] Eloise Matheson, Riccardo Minto, Emanuele GG Zampieri, Maurizio Faccio, and Giulio Rosati. Human–robot collaboration in manufacturing applications: a review. *Robotics*, 8(4):100, 2019.

[Mov67] British Movietone. Unimate - robot, 1967. URL: `https://www.youtube.com/watch?v=hxsWeVtb-JQ`.

[MPW15] Christoph Meier, Alexander Popp, and Wolfgang A Wall. A locking-free finite element formulation and reduced models for geometrically exact kirchhoff rods. *Computer Methods in Applied Mechanics and Engineering*, 290:314–341, 2015.

[MPW19] Christoph Meier, Alexander Popp, and Wolfgang A Wall. Geometrically exact finite element formulations for slender beams: Kirchhoff–love theory versus simo–reissner theory. *Archives of Computational Methods in Engineering*, 26(1):163–243, 2019.

[MSB15] Jörn Malzahn, Russell Schloss, and Torsten Bertram. Link elasticity exploited for payload estimation and force control. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1508–1513. IEEE, 2015.

[NCM$^+$05] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Comparative experiments on task space control with redundancy resolution. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3901–3908. IEEE/RSJ, 2005.

[Nen89] Dragomir N Nenchev. Redundancy resolution through local optimization: A review. *Journal of robotic systems*, 6(6):769–798, 1989.

[NH87] Yoshihiko Nakamura and Hideo Hanafusa. Optimal redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(1):32–42, 1987.

[NKH12] Masoud Nosrati, Ronak Karimi, and Hojat Allah Hasanvand. Investigation of the*(star) search algorithms: Characteristics, methods and approaches. *World Applied Programming*, 2(4):251–256, 2012.

[PA09] Radhakant Padhi and Sk Faruque Ali. An account of chronological developments in control of distributed parameter systems. *Annual Reviews in Control*, 33(1):59–68, 2009.

[Pha14] Quang-Cuong Pham. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 30(6):1533–1540, 2014.

[PKG89] Joey K Parker, Ahmad R Khoogar, and David E Goldberg. Inverse kinematics of redundant robots using genetic algorithms. In *International Conference on Robotics and Automation (ICRA)*, pages 271–276. IEEE, 1989.

[PP18] Hung Pham and Quang-Cuong Pham. Robotic manipulation of a rotating chain. *IEEE Transactions on Robotics*, 34(1):139–150, 2018. doi:10.1109/TRO.2017.2775650.

[PSSB14] Alexander Pekarovskiy, Kunal Saluja, Rohan Sarkar, and Martin Buss. Resonance-driven dynamic manipulation: Dribbling and juggling with elastic beam. In *International Conference on Robotics and Automation (ICRA)*, pages 943–948. IEEE, 2014.

[Ray77] John William Strutt Baron Rayleigh. *The theory of sound*, volume 1. Macmillan, 1877.

[Red13] B Dayanand Reddy. *Introductory functional analysis: with applications to boundary value problems and finite elements*, volume 27. Springer Science & Business Media, New York, USA, 2013.

[Ref13] Reflexxes GmbH. The reflexxes motion libraries, type II. `http://www.reflexxes.ws`, 2013. accessed: Feb. 12, 2016.

[Rei72] Eric Reissner. On one-dimensional finite-strain beam theory: the plane problem. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 23(5):795–804, 1972.

[Rei81] Eric Reissner. On finite deformations of space-curved beams. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 32(6):734–744, 1981.

[RIS13] Zuhayr Rymansaib, Pejman Iravani, and M Necip Sahinkaya. Exponential trajectory generation for point to point motions. In *International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE/ASME, 2013.

[RJCC17] Leonel Rozo, Noémie Jaquier, Sylvain Calinon, and Darwin G Caldwell. Learning manipulability ellipsoids for task compatibility in robot manipulation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3183–3189. IEEE, 2017.

[RLS18] Fabio Ruggiero, Vincenzo Lippiello, and Bruno Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

[RMG18] Alexander Reiter, Andreas Müller, and Hubert Gattringer. On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. *IEEE Transactions on Industrial Informatics*, 14(4):1681–1690, 2018.

[RUC14] Ignacio Romero, Miguel Urrecha, and Christian J Cyron. A torsion-free non-linear beam model. *International Journal of Non-Linear Mechanics*, 58, 2014.

[SBDS02] Ernesto Staffetti, Herman Bruyninckx, and Joris De Schutter. On the invariance of manipulability indices. In *Advances in Robot Kinematics*, pages 57–66. Springer, 2002.

[SGB15] Markus M Schill, Felix Gruber, and Martin Buss. Quasi-direct nonprehensile catching with uncertain object states. In *International Conference on Robotics and Automation (ICRA)*, pages 2468–2474. IEEE, 2015.

[Sic90] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3:201–212, September 1990.

[Sim85] Juan C Simo. A finite strain beam formulation. the three-dimensional dynamic problem. i. *Computer methods in applied mechanics and engineering*, 49(1):55–70, 1985.

[SKY$^+$08] Masayuki Shimizu, Hiromu Kakuya, Woo-Keun Yoon, Kosei Kitagaki, and Kazuhiro Kosuge. Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution. *IEEE Transactions on Robotics*, 24(5):1131–1142, 2008.

[SL87] S Singh and Ming-Chuan Leu. Optimal trajectory generation for robotic manipulators using dynamic programming. *Journal of Dynamic Systems, Measurement, and Control*, 109(2):88–96, 06 1987. doi:10.1115/1.3143842.

[SM86] Kang Shin and N McKay. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Transactions on Automatic Control*, 31(6):491–500, 1986.

[Sni06] Moshe Sniedovich. Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and cybernetics*, 35(3):599–620, 2006.

[Spo86] Graig A Spolek. The mechanics of flycasting: The flyline. *American Journal of Physics*, 54(9):832–836, 1986.

[SRG$^+$16] Nikhil Somani, Markus Rickert, Andre Gaschler, Caixia Cai, Alexander Perzylo, and Alois Knoll. Task level robot programming using prioritized non-linear inequality constraints. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 430–437. IEEE, 2016.

[SS88] Lorenzo Sciavicco and Bruno Siciliano. A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal on Robotics and Automation*, 4(4):403–410, 1988.

[SUO$^+$12] Emre Sariyildiz, Kemal Ucak, Gulay Oke, Hakan Temeltas, and Kouhei Ohnishi. Support vector regression based inverse kinematic modeling for a 7-dof redundant robot arm. In *International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2012.

[SWUL17] Jian Shi, J Zachary Woodruff, Paul B Umbanhowar, and Kevin M Lynch. Dynamic in-hand sliding manipulation. *IEEE Transactions on Robotics*, 33(4):778–795, 2017.

[SWW$^+$19] Philipp S Schmitt, Florian Wirnshofer, Kai M Wurm, Georg v Wichert, and Wolfram Burgard. Planning reactive manipulation in dynamic environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 136–143. IEEE, 2019.

[Tav15] Ali Tavasoli. Dynamic modeling and nonlinear boundary control of hybrid Euler–Bernoulli beam system with a tip mass. *Journal of Multi-body Dynamics*, 229(1):3–15, 2015.

[Tim16] Stephen P Timoshenko. *A course of elasticity theory. Part 2: rods and plates.* A.E. Collins Publishers, St. Petersburg, 1916.

[Tim21] Stephen P Timoshenko. On the correction for shear of the differential equation for transverse vibrations of prismatic bars. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(245):744–746, 1921.

[TK99] Sreenivas Tejomurtula and Subhash Kak. Inverse kinematics in robotics using neural networks. *Information Sciences*, 116(2-4):147–164, 1999.

[TK07] Tanakorn Tantanawat and Sridhar Kota. Design of compliant mechanisms for minimizing input power in dynamic applications. *Journal of Mechanical Design*, 129(10):1064–1075, 2007.

[Ton06] Bertrand Tondu. A closed-form inverse kinematic modelling of a 7r anthropomorphic upper limb based on a joint parametrization. In *International Conference on Humanoid Robots (Humanoids)*, pages 390–397. IEEE-RAS, 2006.

[VAM+12] Nikolaus Vahrenkamp, Tamim Asfour, Giorgio Metta, Giulio Sandini, and Rüdiger Dillmann. Manipulability analysis. In *International Conference on Humanoid Robots (Humanoids)*, pages 568–573. IEEE-RAS, 2012.

[Vic21] Federico Vicentini. Collaborative robotics: a survey. *ASME Journal of Mechanical Design*, 143(4):040802, 2021.

[VPLS18] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018.

[WE84] William A Wolovich and H Elliott. A computational technique for inverse kinematics. In *Conference on Decision and Control (CDC)*, pages 1359–1363. IEEE, 1984.

[WS16] Kenneth J Waldron and James Schmiedeler. Kinematics. In B. Siciliano and O. Khatib, editors, *Springer handbook of robotics*, pages 11–36. Springer, 2nd edition, 2016.

[WSK16] Michael Watterson, Trey Smith, and Vijay Kumar. Smooth trajectory generation on SE(3) for a free flying space robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[Yos85a] Tsuneo Yoshikawa. Dynamic manipulability of robot manipulators. In *International Conference on Robotics and Automation (ICRA)*, pages 1033–1038. IEEE, 1985.

[Yos85b] Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.

[ZBWH13] Franziska Zacharias, Christoph Borst, Sebastian Wolf, and Gerd Hirzinger. The capability map: A tool to analyze robot arm workspaces. *International Journal of Humanoid Robotics*, 10(4), 2013.

[ZFB94] Dimiter Zlatanov, Robert G Fenton, and Beno Benhabib. Singularity analysis of mechanisms and manipulators via a velocity-equation model of the instantaneous kinematics. In *International Conference on Robotics and Automation (ICRA)*, pages 980–985. IEEE, 1994.

Own Thesis-Related Publications

[DOH$^+$15] Khoi Hoang Dinh, Ozgur Oguz, Gerold Huber, Volker Gabler, and Dirk Wollherr. An approach to integrate human motion prediction into local obstacle avoidance in close human-robot collaboration. In *International Workshop on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, 2015. `doi:10.1109/ARSO.2015.7428221`.

[GSH$^+$17] Volker Gabler, Tim Stahl, Gerold Huber, Ozgur Oguz, and Dirk Wollherr. A game-theoretic approach for adaptive action selection in close distance human-robot-collaboration. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

[HGW17] Gerold Huber, Volker Gabler, and Dirk Wollherr. An online trajectory generator on SE(3) with magnitude constraints. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2017.

[Hub19] Gerold Huber. Verfahren zum Regeln eines Roboters mit einem simulierten Roboter, 2019. *European Patent Office,* Publication No. EP3741515.

[HW16] Gerold Huber and Dirk Wollherr. Exploiting elastic dynamics in trajectory optimization. In *International Conference on Robotics and Automation (ICRA), Workshop on Exploiting contact and dynamics in manipulation*. IEEE, 2016.

[HW19] Gerold Huber and Dirk Wollherr. Efficient closed-form task space manipulability for a 7-DOF serial robot. *Robotics*, 8(4):98, 2019. `doi:10.3390/robotics8040098`.

[HW20] Gerold Huber and Dirk Wollherr. An online trajectory generator on SE(3) for human–robot collaboration. *Robotica*, 38(10):1756–1777, 2020. `doi:10.1017/S0263574719001619`.

[HW21] Gerold Huber and Dirk Wollherr. Globally optimal online redundancy resolution for serial 7-dof kinematics along SE(3) trajectories. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[HWB21] Gerold Huber, Dirk Wollherr, and Martin Buss. A concise and geometrically exact planar beam model for arbitrarily large elastic deformation dynamics. *Frontiers in Robotics and AI*, 7:226, 2021. `doi:10.3389/frobt.2020.609478`.

[OGH$^+$16] Ozgur S. Oguz, Volker Gabler, Gerold Huber, Zhehua Zhou, and Dirk Wollherr. Hybrid human motion prediction for action selection within human-robot collaboration. In *International Symposium on Experimental Robotics (ISER)*. IEEE, 2016.