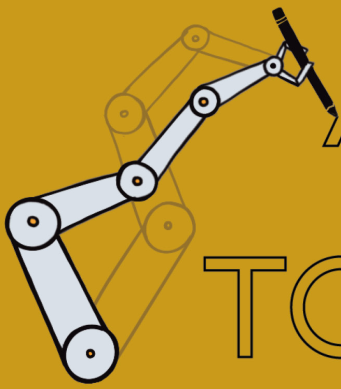
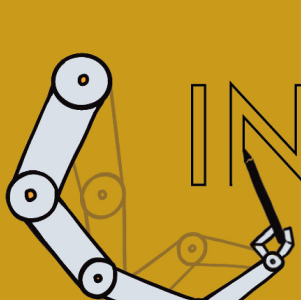
A GEOMETRIC APPROACH TO ROBOTIC MANIPULATION IN PHYSICAL HUMAN-ROBOT INTERACTION





JOHANNES LACHNER

A GEOMETRIC APPROACH TO ROBOTIC MANIPULATION IN PHYSICAL HUMAN-ROBOT INTERACTION

DISSERTATION

to obtain
the degree of Doctor at the University of Twente,
on the authority of the rector magnificus,
Prof. Dr. ir. A. Veldkamp,
on account of the decision of the graduation committee,
to be publicly defended
on Wednesday, July 13, 2022 at 10:45 AM

by

Johannes Lachner

born on January 22, 1987
in Dachau, Germany

This dissertation has been approved by:

Prof. Dr. ir. Stefano Stramigioli, *Supervisor*
Prof. Dr. Neville Hogan, *Supervisor*
Dr.-Ing. Felix Allmendinger, *Co-supervisor*

Cover design:	Beverly Hsu, Cambridge, USA
ISBN:	978-90-365-5356-8
DOI:	10.3990/1.9789036553568

Graduation committee:

Chair:

Prof. Dr.	J.N. Kok	University of Twente, NL

Supervisors:

Prof. Dr. ir.	S. Stramigioli	University of Twente, NL
Prof. Dr.	N. Hogan	Massachusetts Institute of Technology (MIT), USA

Co-supervisor:

Dr.-Ing.	F. Allmendinger	KUKA Deutschland GmbH, DE

Committee members:

Prof. Dr. ir.	H. van der Kooij	University of Twente, NL
Prof. Dr. ir.	A. Franchi	University of Twente, NL
Prof. Dr.-Ing.	A. Albu-Schäffer	Technical University of Munich, DE
Prof. Dr.	B. Siciliano	University of Naples Federico II, IT

The research has been conducted at the Robotics and Mechatronics Group, University of Twente, the Corporate Research Department, KUKA Deutschland GmbH, and the Newman Laboratory for Biomechanics and Human Rehabilitation, MIT.

This work was partially funded by the MURAB project (European Unions Horizon 2020 research and innovation program, Grant Agreement No. 688188), by KUKA Deutschland GmbH, and by the Newman Laboratory for Biomechanics and Human Rehabilitation, MIT.

Dedication / Danksagung

My passion for robotics arose during my time at KUKA Systems GmbH, where I had the chance to take part in creating the first robot applications with the LBR iiwa. The more I learned about robotics, the more questions I had and the deeper I wanted to dive into finding the answers. I asked Dr. Rainer Bischoff, formerly Head of KUKA Corporate Research (CR), if he would support my plan of pursuing a PhD, which led to me becoming the "test tube" for a KUKA PhD program. Rainer, thanks for your trust and great support during this journey.

During my first years at CR, I worked for the European research project MURAB, chaired by the Robotics and Mechatronics (RaM) group of Prof. Dr. Stefano Stramigioli at the University of Twente. Stefano immediately agreed to be my PhD supervisor. Stefano, thanks for all your support and for teaching me so much about the beauty of geometry. Thanks to the people of the RaM group for the great discussions and the nice time we spent at the MURAB demonstrator.

Stefano introduced me to Prof. Dr. Neville Hogan and I was offered the chance to spend one year at the Newman Laboratory for Biomechanics and Human Rehabilitation at MIT. Eventually, Prof. Hogan became my additional PhD supervisor, next to Stefano. Prof. Hogan, thanks for all your precious time and for teaching me about human neuro-motor control, which became my main inspiration for robot control with dynamic interaction. I'm honored to be able working with you in the future. Thanks to the members of the Newman lab, I will never forget the fun we had in and out of the lab.

Most of my research was done at CR. The person with the biggest impact on my PhD is Dr. Felix Allmendinger. As my KUKA supervisor, Felix spent a lot of time with me discussing robotic fundamentals, teaching me how to write scientific papers, and how to perform research in general. Felix, you are not only a mentor but also a true friend. Thanks for all your efforts. Another big thanks goes to the members of CR. You are great experts in your fields and—most of all—very nice people. I will miss working with you!

Spare time was rare during the last few years. However, I remember so many awesome moments with my friends. Eff and Fabi, thanks for being just the best friends. I really appreciate that you always have an open ear and listen to my robot stuff. *Wei'za a Herz habts wia a Bergwerk.* Beverly, thanks for all the help and the nice time we spent in Cambridge, e.g., making Pfannkuchen.

The biggest thanks goes to my family: Mama, Papa, Katharina, and my wife Nathalie. *Mama &Papa, danke für eure bedingungslose Unterstützung. Ohne Euch wäre ich nicht da, wo ich heute bin.* Nathalie, thanks for the indescribable amount of support. You are the love of my life.

Zusammenfassung

Wenn man an Roboter denkt, denkt man an menschenähnliche Maschinen mit sensomotorischen Fähigkeiten. Die Realität sieht jedoch anders aus. In der Industrie werden meist Knickarm-Roboter mit sechs seriell-angeordneten Gelenken verwendet. Diese Roboter werden isoliert von der Umgebung hinter Schutzzäunen platziert. Die Hauptanforderungen an industrielle Roboter sind hohe Geschwindkeit, Präzision und hohe Wiederholgenauigkeit. Zur Beschreibung der Roboterstruktur und zum Programmieren der Roboterbewegung wird eine Vielzahl an Koordinatensystemen genutzt. Die Umgebung des Roboters wird durch externe Sensoren erfasst und die Bauteile werden positionsgetreu platziert.

Im letzten Jahrzehnt wurde eine neue Kategorie von Robotern entwickelt. Die Hauptaufgabe dieser Roboter ist dem Menschen bei schweren und monotonen Arbeiten zu unterstützen. Daher werden diese Roboter meist "Kollaborative Roboter" genannt. Die meist benutzte Terminologie in der Wissenschaft ist "Mensch-Roboter-Interaktion". Da Kollaborative Roboter außerhalb des Schutzzauns agieren, besitzen sie Sensoren, um einen Kontakt mit der Umwelt (z.B. dem Menschen) zu erkennen. Oft sind Kollaborative Roboter kinematisch redundant, was bedeutet, dass der Roboter unendlich viele Achsbewegungen nutzen kann um eine vorbestimmte Werkzeugbewegung auszuführen. Durch die kinematische Redundanz kann ein Roboter oft mehrere Aufgaben gleichzeitig ausführen.

Verglichen mit konventionellen Industrierobotern ist das Programmieren der Roboterapplikation für die Mensch-Roboter-Interaktion komplizierter. Der Programmierer kann sich nicht nur auf den Roboterprozess konzentrieren, sondern muss eine sichere Roboterbewegung gewährleisten. "Sicher" im Kontext der Mensch-Roboter-Interaktion bedeutet, dass Klemmgefahren verhindert und Kontaktkräfte mit dem Menschen minimiert werden müssen. Da Kollaborative Roboter oft eine höhere Anzahl an Freiheitsgraden besitzen als konventionelle Industrieroboter, ist es vermeindlich leichter eine passende Roboterbewegung zu finden. In der Praxis hat sich jedoch die Art der Programmierung nicht geändert. Obwohl der Roboter durch seine kinematische Redundanz theoretisch mehrere Aufgaben gleichzeitig bewältigen könnte, wird üblicherweise nur eine prozessspezifische Aufgabe programmiert. Zudem scheitern traditionelle Regelungsalgorithmen häufig bei der Bewegungsberechnung, da die Abbildung von Werkzeugbewegung auf Achsbewegungen nicht mehr eindeutig ist.

Die vorliegende Arbeit fokusiert sich auf grundlegende Themen der Robotermodellierung und -regelung, mit Schwerpunkt auf Mensch-Roboter-Interaktion. Die Arbeit stellt eine Methodik zur Analyse der Manipulierbarkeit des Roboters bereit. Diese Methodik wird erweitert, um zu untersuchen ob ein Roboter auch mehrere Aufgaben gleichzeitig bewältigen kann. Die Regelungsalgorithmen dieser Arbeit erleichtern die Programmierung von Roboterapplikationen für die Mensch-Roboter-Interaktion. Ein weiterer Fokus der Arbeit liegt auf der Stabilität des Robotersystems während der Regelung.

Die Hauptmethodik der vorliegenden Arbeit ist Differentialgeometrie. Die Arbeit versucht nicht nur die kinematischen Grundlagen für die Bewegungsmodellierung von Robotern zu vermitteln, sondern soll auch zeigen wie die Theorie in der Praxis auf einem Roboter implementiert werden kann. In den entwickelten Analysemethodiken wird die Koordinatenabhängigkeit minimiert. Dies ermöglicht es den Anwendern der Methodiken ihre Analyseergebnisse zu vergleichen. Außerdem werden weniger erfahrene Roboterprogrammierer befähigt, da die Regelungsparameter während der Laufzeit optimiert werden.

Summary

If people think about robots, they think of human-like machines with equivalent cognitive abilities and sensorimotor skills. The reality looks quite different. Most commonly, robots with six serially arranged joints are installed in industrial production. They are placed behind safety fences, isolated from their environment. The main requirements for those robots are fast speed, high accuracy, and repeatability. Coordinates are used to describe the kinematic structure and to program the robot motion. The workpiece is placed as accurately as possible and external sensors are used to capture the environment.

In the last decade, a new sort of robot has evolved. The main purpose of these robots is to assist the human co-worker, e.g., to take over dull and heavy tasks. These robots are often called "collaborative robots." The most common associated terminology in research is "physical Human-Robot Interaction." To ensure the safety of the human co-worker, collaborative robots are equipped with safety features, i.e., sensors to detect contact with the environment. Collaborative robots usually possess more degrees of freedom than conventional robots do. Most common are seven joints, which makes them kinematically redundant. This means that the robot can use infinitely many different joint motions to achieve a desired tool motion.

Compared to conventional industrial applications, the programming of collaborative robot cells is more complicated since the programmer not only has to focus on the robot process, but also has to provide a safe and stable robot motion that does not pose collision and clamping hazards to the human co-worker. Since the robot is more dexterous and can perform multiple tasks, this is supposed to be easier in theory. In practice, however, the programming procedure did not change compared to conventional robots. In most cases, it is only possible to assign one main task to the robot and the kinematic redundancy is not used to manage additional subordinate tasks simultaneously. Even worse, the kinematic redundancy impedes robot programming since an unique relation between tool motion and joint motion does not exist anymore.

This thesis focuses on fundamental topics of robotic manipulation in physical Human-Robot Interaction. Contributions on dexterity, stability, and safety are presented. The thesis proposes a method to analyze the ability of a robot to perform a task. The method is extended to analyze multi-task control. Control approaches are developed that facilitate and speed up the programming of applications in physical Human-Robot Interaction.

The key method of this thesis is differential geometry. Hereby, the thesis tries to help understanding the kinematic foundations and demonstrates how to apply

the theory to a real robot. For users of the presented methods, the results are comparable since the coordinate dependency is kept at a minimum. Moreover, less experienced robot programmers are enabled since the control approaches auto-tune the control parameters during run-time.

Contents

CHAPTER 1

General Introduction

1.1 Motivation

The main requirements for industrial robots are low cycle times, repeatability, and high process quality. Since these robots operate at maximum speed, safety fences are used to protect the human co-worker from possible collisions with the robot. If a person enters the robot cell, the robot motion is stopped as fast as possible and the robot brakes are activated.

For the programming of industrial robots, usually coordinate frames are used. During the process, the robot aligns a body-fixed coordinate frame (e.g., placed on the end-effector) with a world-fixed coordinate frame (e.g., placed on the workpiece). Additionally, the programmer decides how the robot end-effector should move towards the workpiece, e.g., in a linear, point-to-point or circular motion. Once all coordinate frames have been placed, the robot program is transferred to the controller. The Cartesian distance between the coordinate frames is interpolated and a feasible joint configuration for each Cartesian position is computed. This is usually done by solving the inverse kinematic problem for each Cartesian position. Lastly, the resulting joint trajectory is post-processed in order to achieve a robot motion with high performance (e.g., high Cartesian position accuracy, fast joint motion, or low jerk). For the robot process, positioning mechanisms and external sensors are used to ensure that the workpiece is placed in the exact position.

During recent years, a new generation of robots has been developed. Such collaborative robots are often placed next to human co-workers. For this reason, those robots need to incorporate safety features and guarantee stability in order to enable an operation without a safety fence. For example, torque sensors incorporated in the robot structure are used to detect collisions with the environment along the whole robotic arm. Since the human co-worker and the robot share a workspace, the safety certification of collaborative robots is very complicated. All risks sources have to be identified and measures provided to minimize the risks for the human co-worker.

The torque sensors can additionally be used to “sense” the environment. Costly position mechanisms and external sensors can hence be saved. However, the robot process is still dependent on a deliberate placement of coordinate frames. Moreover, the desired trajectory is calculated offline and cannot be modified during motion execution. If a collaborative robot is kinematically redundant, more advanced control is needed since infinitely many sets of joint motions can generate a desired workspace motion.

The goal of industrial collaborative robots is to automate manual processes that are usually done by humans, e.g., assembly processes. Humans are capable of managing multiple tasks simultaneously. These tasks are not only positioning tasks as for most industrial robots, but also involve interaction with the environment. The human task performance is enabled by cognition: We perceive our environment with our eyes and interact with it through our sensorimotor skills. Hereby, humans often manage to do repetitive tasks without even looking, e.g., during assembly line work. These are suitable applications for collaborative robots with "sensing abilities," e.g., robots with integrated torque sensors.

As collaborative robots are interacting with the environment, they should be treated as part of the surrounding physics. To enable this goal, some of the fundamentals of robotic manipulation have to be approached. This thesis will make contributions to robotic dexterity, stability, and safety.

1.2 Research questions of this thesis

For industrial robotics, it is important to have a measure of "how well" a robot can perform a given task. This measure is already needed during the design phase of the kinematic structure since different joint type choices will lead to different dexterity results, e.g., through combining linear and rotational joints. Moreover, for a given customer task, the measure can help to select the right robot. Lastly, such a measure can be applied online during the operation of the robot and can indicate that the robot cannot perform its task. This leads to the first research question. *Q1: Can we define dexterity of robotic manipulators in a coordinate-invariant way?* (→ chapter 2)

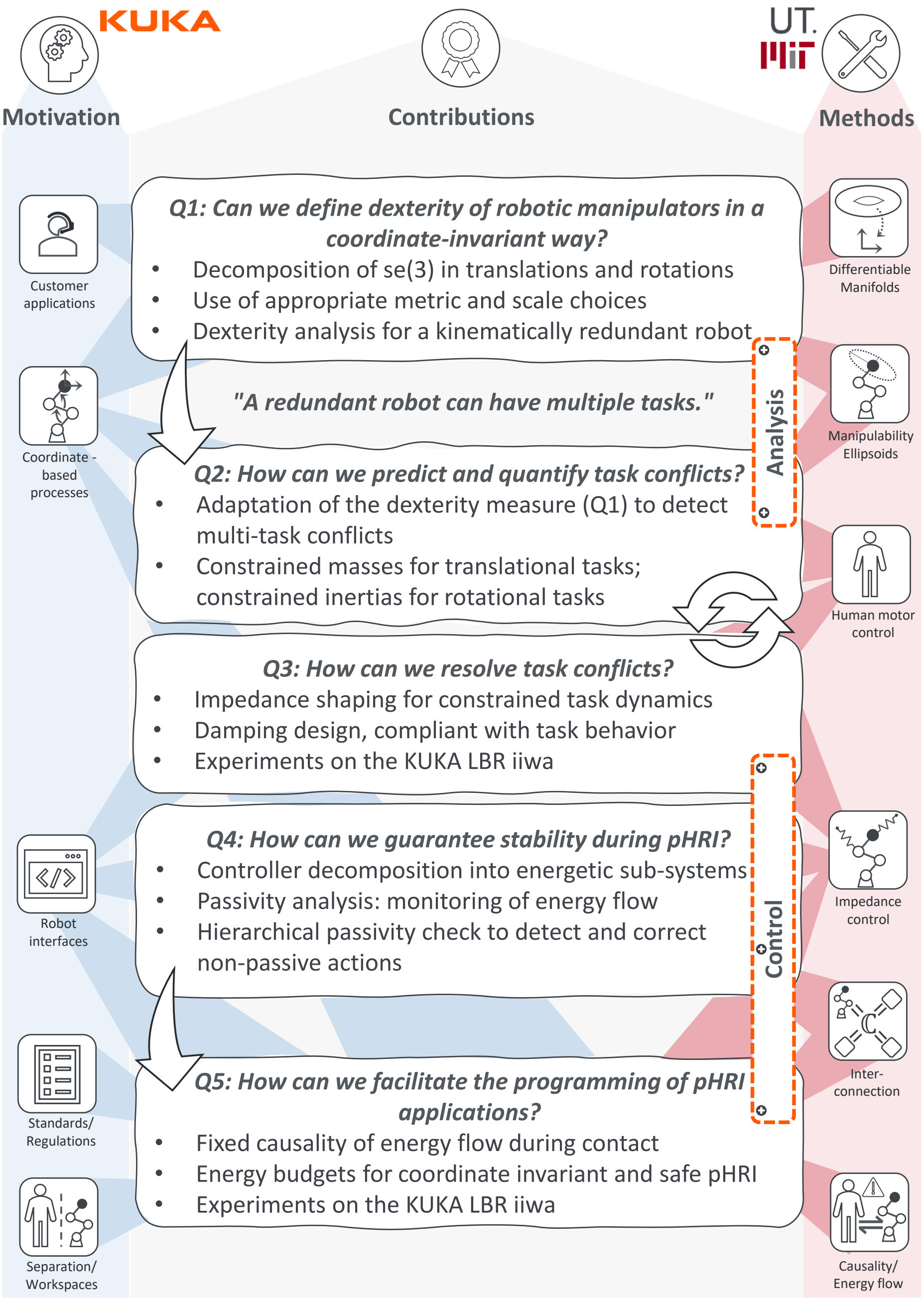
The dexterity of robotic manipulators is linked to kinematic singularities: The robot cannot move along/about singular directions. For an impedance controlled robot, the robot programmer expresses the desired dynamic task behavior. Near singular configurations, the robot is not able to perform the desired task dynamics. To be productive, industrial applications always assign one main task to the robot. However, a kinematically redundant robot can perform multiple tasks simultaneously. The kinematic redundancy can be used to optimize the robot configuration, as well as the kinematic and dynamic properties during the process. Since the main task is essential for the robotic process, it is important to detect and quantify a conflict between the main task and subordinate tasks. This leads to the second research question. *Q2: How can we predict and quantify task conflicts?* (→ chapter 3)

With simple impedance superposition, the control of multiple tasks is straight forward since different impedances can be superimposed. Through superposition, all tasks have the same priority level. For industrial applications, however, the main tasks must not be influenced by tasks with lower priority. This leads to the third research question. *Q3: How can we resolve task conflicts for multi-task impedance control?* (→ chapter 3)

Due to its stability properties, an impedance controller is a beneficial control approach for industrial applications that involve contact with the environment, e.g., assembly applications. However, an online modification of the control parameters can violate the passivity properties of the controller. For industrial robots, a threshold for the commanded motor current or motor torque is typically set. If the control command would violate the threshold, the robot motion is stopped as fast as possible and the brakes are activated. Hence, clamping scenarios can arise, which impede safe physical Human-Robot Interaction (pHRI). This leads to the fourth research question. *Q4: How can we guarantee stability during pHRI?* (→ chapter 3)

All industrial applications with pHRI have to be certified by the producer of the robot cell. The risk assessment and risk minimization compromises a high percentage of the certification process. Therefore, pHRI-cells are more expensive than traditional robot cells. Two major risks are collisions and clamping scenarios. As for the robot programming, the safety implementation of the application also has strong dependencies on coordinates, e.g., to monitor the robot velocity or to predict external forces. However, the current certification process does not in general guarantee a safe robot behavior. It is not possible to predict all risks in advance, especially in unstructured environments. Moreover, multiple control parameters have to be adapted during the certification process, which requires experienced robot programmers. This leads to the fifth research question. *Q5: How can we facilitate the programming of pHRI applications?* (→ chapter 4)

1.3 Thesis overview



Q1: Can we define dexterity of robotic manipulators in a coordinate-invariant way?

- Decomposition of se(3) in translations and rotations
- Use of appropriate metric and scale choices
- Dexterity analysis for a kinematically redundant robot

"A redundant robot can have multiple tasks."

Q2: How can we predict and quantify task conflicts?

- Adaptation of the dexterity measure (Q1) to detect multi-task conflicts
- Constrained masses for translational tasks; constrained inertias for rotational tasks

Q3: How can we resolve task conflicts?

- Impedance shaping for constrained task dynamics
- Damping design, compliant with task behavior
- Experiments on the KUKA LBR iiwa

Q4: How can we guarantee stability during pHRI?

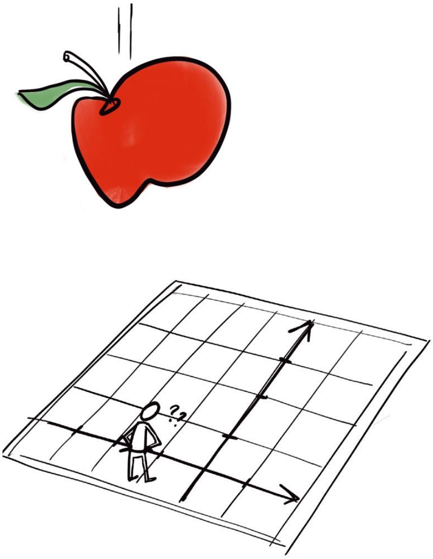
- Controller decomposition into energetic sub-systems
- Passivity analysis: monitoring of energy flow
- Hierarchical passivity check to detect and correct non-passive actions

Q5: How can we facilitate the programming of pHRI applications?

- Fixed causality of energy flow during contact
- Energy budgets for coordinate invariant and safe pHRI
- Experiments on the KUKA LBR iiwa

CHAPTER 2

The influence of coordinates in robotic dexterity analysis



Lachner J, Schettino V, Allmendinger F, Fiore MD, Ficuciello F, Siciliano B and Stramigioli S (2020) The influence of coordinates in robotic manipulability analysis. *Mechanism and Machine Theory* 146. doi: 10.1016/j.mechmachtheory.2019.103722

2.1 Introduction

2.1.1 Kinematic foundations

The set of all robot configurations $\boldsymbol{q}$ can be associated to the manifold Q. The set of all homogeneous transformations $\boldsymbol{H}$ constitutes the manifold $SE(3)$. The robot workspace W can be associated to a sub-manifold of $SE(3)$. Both manifolds can be seen in fig. 2.1.

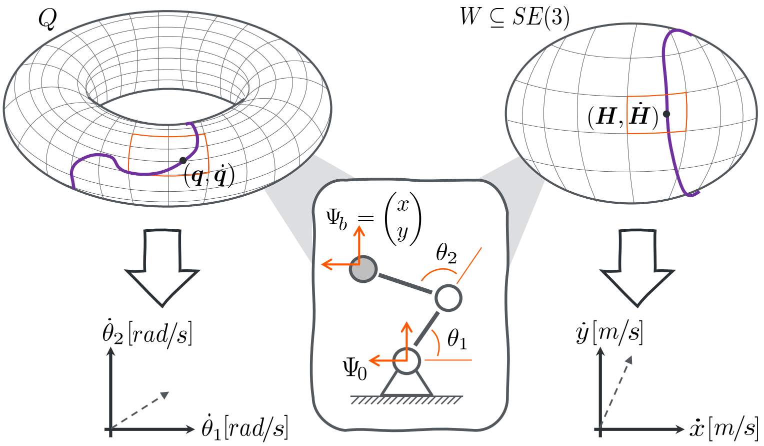

Q
$W \subseteq SE(3)$
$(\boldsymbol{q}, \dot{\boldsymbol{q}})$
$(\boldsymbol{H}, \dot{\boldsymbol{H}})$
$\Psi_b = \begin{pmatrix} x \\ y \end{pmatrix}$
θ_2
θ_1
Ψ_0
$\dot{\theta}_2\,[rad/s]$
$\dot{\theta}_1[rad/s]$
$\dot{y}\,[m/s]$
$\dot{x}\,[m/s]$

Figure 2.1: Two differential manifolds in robotics: The set of all joint configurations form the manifold Q and the set of all homogeneous transformations form the manifold $SE(3)$. Charted elements of Q form the joint space and are locally isomorphic to $\mathbb{R}^n$. Charted elements of W form the workspace and are locally isomorphic to $\mathbb{R}^6$.

The trajectory of the robot can be represented as a smooth curve on Q and the trajectory of the end-effector as a smooth curve on $W \subseteq SE(3)$. If the curve parameter is time $t \in \mathbb{R}$, the instantaneous robot configuration and the instantaneous homogeneous transformation are $\boldsymbol{q}(t) \in Q$ and $\boldsymbol{H}(t) \in SE(3)$, respectively. If the end-effector behavior is the main interest, the programmer of the robot wants to describe the workspace trajectory $\boldsymbol{H}(t)$. For position-controlled robots, the task of the robot controller is to transform $\boldsymbol{H}(t)$ to a feasible joint trajectory $\boldsymbol{q}(t)$ which can be directly controlled.

The tangent space to the smooth manifold Q at joint configuration $\boldsymbol{q}$ is denoted $T_{\boldsymbol{q}}Q$. Since $SE(3)$ is not only a smooth manifold but also a group, $SE(3)$ is called a “Lie group.” The tangent space to the identity element e of the group is denoted $T_eSE(3)$. For $SE(3)$, the identity element is the 4×4 identity matrix. Elements of $T_{\boldsymbol{q}}Q$ and $T_eSE(3)$ are called “tangent vectors.”

$T_eSE(3)$ has the structure of a Lie algebra, indicated as $se(3)$. The properties of $se(3)$ are shown in appendix B and will be analyzed in more detail in chapter 2.3. $T_eSE(3)$ has the structure of a vector space with three operations: vector addition, scalar multiplication, and vector product. Vector addition and scalar multiplication are part of the vector space structure and multiplication exists since $T_eSE(3)$ is a Lie algebra. The space of linear maps from $T_eSE(3)$ to real numbers is called the "co-tangent space" $T_e^\star SE(3)$. Also $T_e^\star SE(3)$ has the structure of a Lie algebra, indicated as $se^\star(3)$. Moreover, $T_e^\star SE(3)$ is a vector space which is called the "co-vector space."

Coordinates result from a bijective map that assigns a tuple of real numbers to a point on the manifold. Joint coordinates are a map of a point on Q and task coordinates are a map of a point on $SE(3)$. The parameterization of joint coordinates is dependent on the type of joints, e.g., angles in $[rad]$ are mostly used for revolute joints and distances in $[m]$ are mostly used for linear joints. Task coordinates can be associated to Cartesian coordinate frames. A Cartesian coordinate frame is a set of three mutually orthonormal vectors with unit length. The coordinate frame can either be attached to the moving robot body or placed stationary in the workspace. The former is called a "body-fixed coordinate frame" and the latter is called an "inertial coordinate frame."

For each time instant, the current point on $SE(3)$ represents a change of motion from a reference coordinate frame (body-fixed or inertial) to the current value $\boldsymbol{H}(t) \in SE(3)$. $\boldsymbol{H}(t)$ incorporates translations $\boldsymbol{p}(t) \in \mathbb{R}^3$ and rotations $\boldsymbol{R}(t) \in SO(3)$. Translations $\boldsymbol{p}(t)$ can be represented by the Cartesian coordinate frame with axes unit $[m]$.

Rotations are elements of smooth manifolds and can be represented globally and locally. For global representations, there is a bijective relation between all elements of the manifold and their coordinate representation. The matrix $\boldsymbol{R}(t)$ is a global representation of $SO(3)$ with nine-parameters. For local representations, the mapping only exists for a subset of rotations. Moreover, representation singularities can occur. In robotics, often Euler angles are used, which are local representations with three parameters. A comparison of all rotation representations can be found in Shuster (1993).

One can find a local coordinate representation of tangent vectors with respect to the coordinate-induced bases. Tangent vectors of $T_{\boldsymbol{q}}Q$ are locally isomorphic to $\mathbb{R}^n$, with n being the number of robot joints. The local coordinate representation of $T_{\boldsymbol{q}}Q$ yields the joint velocity $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ of the robot. Tangent vectors of $T_eSE(3)$ are called "twists." A twist can either be expressed in a fixed inertial coordinate frame (called "spatial twist") or in a moving body-fixed coordinate frame (called "body twist"). The spatial twist is calculated by $\dot{\boldsymbol{H}}\boldsymbol{H}^{-1} \in se(3)$ and the body twist is calculated by $\boldsymbol{H}^{-1}\dot{\boldsymbol{H}} \in se(3)$. Here, the notation of the dependency on t was dropped for simplicity. Tangent vectors of $T_eSE(3)$ are locally isomorphic to $\mathbb{R}^6$. If $T_eSE(3)$ is associated with $\mathbb{R}^6$, the 0-element of the Lie algebra is the zero vector. The twists $\dot{\boldsymbol{H}}\boldsymbol{H}^{-1}$ and $\boldsymbol{H}^{-1}\dot{\boldsymbol{H}}$ can be represented as six-dimensional vectors. The vector components then correspond to the standard basis of $se(3)$ (Zefran and Kumar, 1997; Zefran et al., 1999). In this work $\boldsymbol{\xi}^{\mathrm{s}} \in \mathbb{R}^6$ will be denoted as the vector form of the spatial twist $\dot{\boldsymbol{H}}\boldsymbol{H}^{-1}$ and $\boldsymbol{\xi}^{\mathrm{b}} \in \mathbb{R}^6$ will be denoted as the vector form of the body twist $\boldsymbol{H}^{-1}\dot{\boldsymbol{H}}$. The coordinate dependency of both twists will be shown in more detail in the next subchapter.

2.1.2 Left and right invariant representation of twists

The position of a point expressed in the body-fixed coordinate frame “b” is denoted ${}^{\mathrm{b}}\boldsymbol{p} \in \mathbb{R}^3$. This point, written in homogeneous coordinates, can be expressed with respect to the inertial coordinate frame “0” by using the homogeneous transformation matrix ${}^{0}\boldsymbol{H}_{\mathrm{b}} \in SE(3)$:

$$\begin{pmatrix} {}^{0}\boldsymbol{p} \\ 1 \end{pmatrix} = {}^{0}\boldsymbol{H}_{\mathrm{b}} \begin{pmatrix} {}^{\mathrm{b}}\boldsymbol{p} \\ 1 \end{pmatrix} . \tag{2.1}$$

For a fixed point on the robot body, time differentiation of eq. (2.1) yields:

$$\begin{pmatrix} {}^{0}\dot{\boldsymbol{p}} \\ 1 \end{pmatrix} = {}^{0}\dot{\boldsymbol{H}}_{\mathrm{b}} \begin{pmatrix} {}^{\mathrm{b}}\boldsymbol{p} \\ 1 \end{pmatrix} . \tag{2.2}$$

Here, ${}^{0}\dot{\boldsymbol{H}}_{\mathrm{b}}(t) \in T_{H(t)}SE(3)$ is the tangent vector to a point on the workspace trajectory. In eq. (2.2), a dependency on “b” and “0” still exists. To resolve the dependency on the body-fixed coordinate frame, the point ${}^{\mathrm{b}}\boldsymbol{p}$ (in homogeneous coordinates) can be transformed to inertial coordinates:

$$\begin{pmatrix} {}^{0}\dot{\boldsymbol{p}} \\ 1 \end{pmatrix} = \underbrace{{}^{0}\dot{\boldsymbol{H}}_{\mathrm{b}} \left({}^{0}\boldsymbol{H}_{\mathrm{b}}\right)^{-1}}_{\text{Spatial twist}} \begin{pmatrix} {}^{0}\boldsymbol{p} \\ 1 \end{pmatrix} . \tag{2.3}$$

This yields the spatial twist $\dot{\boldsymbol{H}}\boldsymbol{H}^{-1} \in T_eSE(3)$, which is derived by “right translation” (Park et al., 1993) to the identity of the group, i.e., the matrix $\boldsymbol{H}^{-1}$ is multiplied from the right. Here, the notion of the dependency on coordinates was dropped for simplicity. As can be seen in eq. (2.3), the spatial twist does not depend on the choice for a body-fixed coordinate frame. Hence, the spatial twist is called a “right invariant representation” of $\dot{\boldsymbol{H}}$.

To resolve the dependency on the inertial coordinate frame, eq. (2.2) can be multiplied by $\left({}^{0}\boldsymbol{H}_{\mathrm{b}}\right)^{-1}$ from the left:

$$\begin{pmatrix} {}^{\mathrm{b}}\dot{\boldsymbol{p}} \\ 1 \end{pmatrix} = \underbrace{\left({}^{0}\boldsymbol{H}_{\mathrm{b}}\right)^{-1} {}^{0}\dot{\boldsymbol{H}}_{\mathrm{b}}}_{\text{Body twist}} \begin{pmatrix} {}^{\mathrm{b}}\boldsymbol{p} \\ 1 \end{pmatrix} . \tag{2.4}$$

This yields the body twist $\boldsymbol{H}^{-1}\dot{\boldsymbol{H}} \in T_eSE(3)$, which is derived by “left translation” (Park et al., 1993) to the identity of the group, i.e., the matrix $\boldsymbol{H}^{-1}$ is multiplied from the left. Again, the notion of the dependency on coordinates was dropped for simplicity. The body twist is called a “left invariant representation” of $\dot{\boldsymbol{H}}$.

2.1.3 Metric property of SE(3)

In robotics, it is often desired to talk about the length[1] or the norm of a vector. However, the scalar product structure of $\mathbb{R}^3$ cannot be generalized for $T_eSE(3)$.

[1] To talk about the length of a vector, an inner product has to be defined first.

Since $T_eSE(3)$ can be associated with $\mathbb{R}^6$, it incorporates a (pseudo-)inner product structure with a quadratic form $g \in SE(3)$. If g is non-degenerate, it is called a “pseudo-metric” and if g is positive definite it is called a “metric.”

Definition 2.1 *Given a $\mathbb{R}$-vector space V, a pseudo-inner product with bi-linear pseudo-metric g will be defined by imposing two conditions on g:*

1. *Symmetry:* $\forall v, w \in V : g(v,w) = g(w,v)$
2. *Non-degeneracy:*

$$\begin{gathered} \tilde{g} : V \to V^\star \\ v \mapsto \tilde{g}(v) \\ \exists\ \tilde{g}^{-1} : V^\star \to V \\ (v,w) \mapsto g(v,w) := \tilde{g}(v)(w) \ \textit{and}\ \tilde{g}^{-1} \circ g = id_V \end{gathered}$$

A stronger requirement than non-degeneracy is positive definiteness, which yields an inner product with metric $\bar{g}$:

$$\bar{g}(v,v) \geq 0 \ \textit{and}\ \bar{g}(v,v) = 0 \Rightarrow v = 0$$

As shown in sec. 2.1.2, the twists $\in T_eSE(3)$ can either be derived by right or left translation of $\dot{\boldsymbol{H}}(t) \in T_{H(t)}SE(3)$ to the identity element of the group. Hence, the (pseudo-)inner product on $se(3)$ will also have two classes of (pseudo-)metrics: right and left invariant (pseudo-)metrics. A (pseudo-)metric that is right and left invariant is called “bi-invariant.”

As shown in Murray et al. (1994), any bi-invariant quadratic form in $se(3)$ can be represented by a linear combination of the “Killing form” and the “Hyperbolic form”:

$$\mathscr{M} = \underbrace{\gamma \begin{pmatrix} 0 & 0 \\ 0 & \boldsymbol{I} \end{pmatrix}}_{\text{Killing form}} + \underbrace{\beta \begin{pmatrix} 0 & \boldsymbol{I} \\ \boldsymbol{I} & 0 \end{pmatrix}}_{\text{Hyperbolic form}} . \tag{2.5}$$

Here, $\boldsymbol{I} \in \mathbb{R}^3$ is the identity matrix and the scalars γ and β are scale choices.

2.1.4 Dexterity and Manipulability

Dexterity analysis makes a statement about how “well” or “badly” the robot can be moved along different workspace trajectories. Since the trajectory is a curve in $SE(3)$ (upper part of fig. 2.1), dexterity is an intrinsic condition of the robot and must be invariant of the coordinate choice of the user. More specifically, the dexterity analysis must be invariant of the chosen joint coordinates to describe the kinematic structure and the task coordinates (body-fixed or inertial) to describe the workspace behavior (lower part of fig. 2.1).

In the literature, dexterity measures have different purposes and should support the user during every phase of the robot’s life cycle. These phases can roughly be divided in:

1. Design phase → 2. Kinematic and Dynamic analysis
→ 3. Task planning → 4. Control and operation.

During the first phase, the designer mostly analyzes the available workspace volume[2] of the robot and checks if the robot can reach a set of goal coordinate frames (Park, 1995). During the kinematic and dynamic analysis, the robot structure is optimized such that the robot is able to produce desired velocities, accelerations, and forces in certain given directions (Zefran et al., 1999). The dynamics of the robot are mostly calculated by placing body-fixed coordinate frames on each link such that the coordinate frame origin coincides with the respective center of mass. Since the designer adapts the kinematic structure during these two phases, the position of the body-fixed coordinate frames changes. Changing the inertial coordinate frame, however, should not change the results of the robot dynamic calculation. It is therefore desired that the dexterity measure should be invariant of the inertial coordinate frame (left invariant) (Park et al., 1993). The third phase involves the planning of the robot cell. For a given robot task, a dexterity measure should support the user to select the right robot. In most cases, the exact tool dimensions are not known in this phase. Hence, the dexterity measure should be invariant to the body-fixed coordinate frame (right invariant) (Park and Brockett, 1994). For the last phase, a dexterity measure can observe (and at best predict) a bad robot condition, i.e., a kinematic singularity or a task conflict[3].

In the literature, the terminology "manipulability analysis" is often used instead of "dexterity analysis." Hereby, manipulability addresses a part of dexterity. More precisely, manipulability can also be called "local dexterity" since it is applied in the latter three phases of the robot life cycle. The purpose of manipulability measures is to determine if the robot is close to a kinematic singularity (Park and Brockett, 1994).

The main aim of this chapter is to show the relations between dexterity analysis and the choice of both coordinate types, i.e., joint coordinates and task coordinates (fig. 2.1). In a first step, the heavily used "Velocity Manipulability Ellipsoid" (Yoshikawa, 1985b) is analyzed and it is shown that different ellipsoids can arise for one and the same robot. The reasons can be illustrated with the help of tensor analysis. In a further step, a dexterity measure is derived that describes a dynamic characteristic and can be applied to kinematically redundant robots with mixed joint types. For a given control input, the robot end-effector will produce generalized motions that are associated to the robot task. With the help of screws, the end-effector motion can be decomposed into translational and rotational motions. The analysis of the robot's ability to produce end-effector motions has dependencies on task coordinates. It will be shown that for purely rotational end-effector motions, the dependency on task coordinates can be removed. For translational end-effector motions, specific directions have to be chosen for dexterity analysis. Summarized, the main contributions of this chapter are:

- Overview of the most frequently used dexterity measures in the literature.

[2] To talk about a volume, a volume form on a manifold has to be defined first.

[3] It will be shown in chapter 3 that both are strongly related.

- Derivation of the Velocity Manipulability Ellipsoid and identification of coordinate dependencies.
- Analysis of the differential geometric workspace structure to understand the relations to coordinates.
- Derivation and representation of a coordinate invariant dexterity measure for rotational task motions.
- Illustration of possibilities and constraints for the analysis of translational task motions.

The results of this chapter are partly reported in Lachner et al. (2020).

2.2 Dexterity measures in the literature

The manipulability measures in the literature are strongly related to the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}} \in \mathbb{R}^{6\times n}$. $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ maps joint velocities to a six-dimensional vector of task velocities $\boldsymbol{V}$. The first three rows of $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ can be derived by $\partial L(\boldsymbol{q})/\partial \boldsymbol{q}$ and yield the time differentiation of the position $\boldsymbol{p} = L(\boldsymbol{q}) \in \mathbb{R}^3$, i.e., the linear velocity $\dot{\boldsymbol{p}} \in \mathbb{R}^3$. The last three rows can be derived with the time derivative of the rotation matrix and by using the skew-symmetric representation of the angular velocity $\boldsymbol{w} \in \mathbb{R}^3$ (Siciliano et al., 2009):

$$\boldsymbol{V} = \boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}\ \dot{\boldsymbol{q}}. \tag{2.6}$$

$\boldsymbol{V}$ is often called “spatial velocity” but should not be confused with twists. For the derivation of a twist, no time differentiation is needed. Through the analytical derivation of linear velocities and geometrical derivation of angular velocities, $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ will be called “Hybrid Jacobian Matrix.”

A geometric solution of the Jacobian matrix that incorporates the structure of a Lie Group is the matrix $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}} \in \mathbb{R}^{6\times n}$. $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$ maps joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to spatial twists in vector form $\boldsymbol{\xi}^{\mathrm{s}} \in \mathbb{R}^6$:

$$\boldsymbol{\xi}^{\mathrm{s}} = \boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}\ \dot{\boldsymbol{q}}. \tag{2.7}$$

The columns of $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$ are the twist axes of the robot joints in the configuration $\boldsymbol{q}$ (Murray et al., 1994). In the following, $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$ will be called “Geometrical Jacobian Matrix.” As shown in sec. 2.1.2, $\boldsymbol{\xi}^{\mathrm{s}}$ is an element of the Lie algebra and can be derived by right translation of $\dot{\boldsymbol{H}}(t)$ to the identity element of the group and the homogeneous transformation $\boldsymbol{H}(t)$ can be integrated from it.

Due to the differentiation of the forward kinematic map $L(\boldsymbol{q})$, $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ of eq.(2.6) has a dependency on the body-fixed coordinate frame *and* the inertial coordinate frame. This must be considered when using $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ for dexterity analysis. For manipulability analysis, $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ was examined in Salisbury and Craig (1982), Paul and Stevenson (1983), and Angeles (2014). For simplicity, $\boldsymbol{J}(\boldsymbol{q}) \coloneqq \boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ in the sequel of chapter 2.2.

A lot of authors have proposed many different dexterity measures (tab. 2.1). As already mentioned, there are different objectives behind the measures: Some focus on the design phase of the robot, some focus on kinematic/dynamic analysis, while others focus on control. While a lot of measures were proposed in the 1980's, the geometric fundamentals of dexterity analysis in $SE(3)$ were reported later in the 1990's. The dexterity measures in the upper part of tab. 2.1 ignore the dependencies on coordinates. The dexterity measures in the lower part consider task coordinates but not the dependency on joint coordinates. Paden and Sastry (1988) derived a bi-invariant dexterity measure. All other dexterity measures in the lower part of the table consider invariance to the inertial coordinate frame (left invariance). Hence, it was identified a lack of dexterity measures for kinematically redundant robots that take into account joint coordinates *and* task coordinates. Moreover, the dexterity measures in the lower part of tab. 2.1 focus on design and kinematic analysis, but not on control. The derived dexterity measure of this chapter will be applied for torque control of a kinematically redundant robot in chapter 3.

Table 2.1: Dexterity measures proposed for different objectives: Design (D), kinematic/dynamic analysis (A), and control (C). The measures in the upper part of the table include a matrix and an operation, performed on that matrix. The abbreviations for the matrix operations are: $cond_2(\bullet)$ is the condition number based on the matrix 2-norm, $det(\bullet)$ the matrix determinant, $eig(\bullet)$ eigenvalue decomposition, and $svd(\bullet)$ singular value decomposition. The work in the lower part of the table reports the geometric fundamentals of dexterity analysis in $SE(3)$.

Author, Year	**Measure**	**D**	**A**	**C**
Salisbury and Craig (1982)	$cond_2(\boldsymbol{J}(\boldsymbol{q}))$		✓	
Paul and Stevenson (1983)	$\lvert det(\boldsymbol{J}(\boldsymbol{q}))\rvert$		✓	
Asada (1983)	$eig((\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T)^{-1})$		✓	
Hogan (1984)	$eig(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T)$		✓	✓
Yoshikawa (1985b)	$\sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}$		✓	
Yoshikawa (1985a)	$det(\boldsymbol{J}(\boldsymbol{q})(\boldsymbol{M}(\boldsymbol{q})^T\boldsymbol{M}(\boldsymbol{q}))^{-1}\boldsymbol{J}(\boldsymbol{q})^T)$		✓	
Ma and Angeles (1991)	$svd(\boldsymbol{J}(\boldsymbol{q})) \rightarrow \frac{\sigma_{\max}}{\sigma_{\min}}$		✓	✓
Paden and Sastry (1988)	Fundamentals of dexterity analysis in $SE(3)$	✓		
Park et al. (1993)		✓		
Park and Brockett (1994)		✓	✓	
Park (1995)		✓		
Zefran et al. (1999)			✓	

2.2.1 Velocity Manipulability Ellipsoid

The quadratic form $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ is probably the most frequently used measure of dexterity. It was originally proposed by Yoshikawa (1985b) and e.g., applied in Klein and Blaho (1987), Baillieul (1987), Chiu (1988), Burget and Bennewitz (2015), and Su et al. (2019). Hereby, $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ is the core of the “Velocity Manipulability Ellipsoid” (Yoshikawa, 1985b), which is often recommended for manipulability analysis of kinematically redundant robots ($n > 6$) (Burget and Bennewitz, 2015; Chen et al., 2015; Duan et al., 2019).

The derivation of this ellipsoid starts by visualizing the velocities $\dot{\boldsymbol{q}}$ as a sphere. The used sphere equation is

$$\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}} = 1. \tag{2.8}$$

As a next step of the derivation, a Jacobian (pseudo-)inverse has to be defined which maps task velocities $\boldsymbol{V}$ to joint velocities $\dot{\boldsymbol{q}}$. If $\boldsymbol{J}(\boldsymbol{q})$ is square and has full rank, the Jacobian inverse is uniquely defined, since the mapping is bijective. In the kinematically redundant case ($n > 6$), the mapping is surjective and hence there are infinitely many solutions of $\dot{\boldsymbol{q}}$ to produce a desired Cartesian velocity. To choose a solution among the possible ones, a choice for a cost function can be made, which is minimized. A typical choice for a cost function is

$$g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{W}\dot{\boldsymbol{q}}, \tag{2.9}$$

with $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ being a symmetric positive definite weighting matrix. The solution obtained by minimizing eq. (2.9) is the “Generalized Inverse” (Siciliano et al., 2009) of $\boldsymbol{J}(\boldsymbol{q})$:

$$\boldsymbol{J}(\boldsymbol{q})^{\#} = \boldsymbol{W}^{-1}\boldsymbol{J}(\boldsymbol{q})^T\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{W}^{-1}\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1}. \tag{2.10}$$

Choosing $\boldsymbol{W}$ equal to the Identity matrix yields the “Moore-Penrose Inverse” (Penrose, 1955) $\boldsymbol{J}(\boldsymbol{q})^{\dagger}$. By using $\boldsymbol{J}(\boldsymbol{q})^{\dagger}$, one obtains the solution $\dot{\boldsymbol{q}}$ that minimizes $g(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}}$. This choice is heavily used to determine joint velocities for a given set of task velocities:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{\dagger}\boldsymbol{V}. \tag{2.11}$$

Substituting eq. (2.11) in eq. (2.8) results in a workspace ellipsoid:

$$\boldsymbol{V}^T\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1}\boldsymbol{V} = 1. \tag{2.12}$$

The shape and orientation of the ellipsoid is determined by the matrix $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$. It will be shown in the next chapter that the choice $\boldsymbol{J}(\boldsymbol{q})^{\dagger}$ has consequences when using eq. (2.12) for the analysis of robots with different joint coordinate units.

2.2.2 Specific aspects of $J(q)J(q)^T$

The quadratic form $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ is incorporated in the manipulability measure (tab. 2.1):

$$\mathbb{R} \ni \mathfrak{w} = \sqrt{det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}, \tag{2.13}$$

where $det(\bullet)$ is the determinant of a matrix. The determinant of a matrix is equal to the product of its eigenvalues. A definition of eigenvalues can be found in

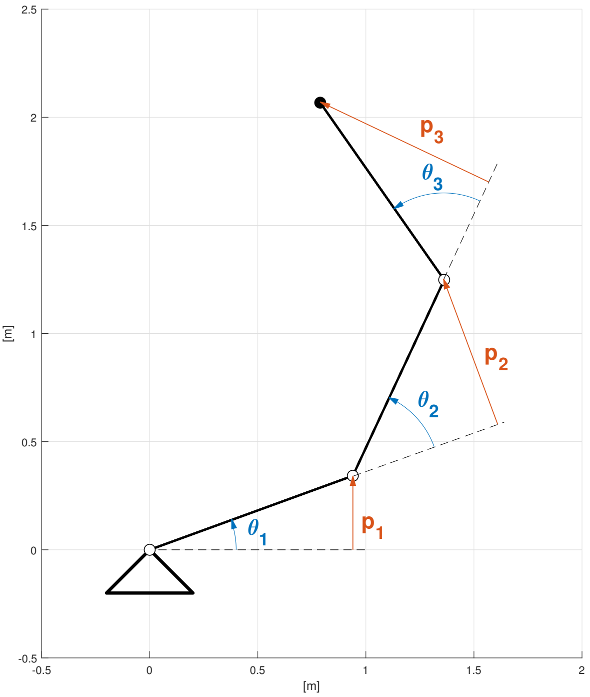


Figure 2.2: 3-DOF planar robot with different set of joint coordinates: blue set $(\theta_1, \theta_2, \theta_3)$ with unit $[rad]$ and red set (p_1, p_2, p_3) with unit $[m]$. The values of the joint coordinate sets can be seen in appendix A.2.

Dubrovin et al. (1984). It will be shown in sec. 2.3.2 that eigenvalues of quadratic forms have dependencies on a chosen (pseudo-)metric.

Since $\mathfrak{w}$ is independent of the choice of the body-fixed coordinate frame (appendix A.1) and can be applied for kinematically redundant robots, it is often recommended for the analysis of mobile robots (Siciliano and Khatib, 2008; Siciliano et al., 2009; Burget and Bennewitz, 2015; Duan et al., 2019).

The matrix $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ has dependencies on joint coordinates. In fig. 2.2, a robot with different sets of joint coordinates is illustrated. Assigning different sets for the same robot yields different values of $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ and in its eigenvalue decomposition. If the set of coordinates is chosen such that not all coordinates bear the same unit, the matrix product $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ even cannot be calculated because a mismatch of units arises (cf. previous work of Schwartz et al. (2002), Merlet (2006), and appendix A.2). This has the consequence that $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ cannot be computed for a mobile robot without separating base and robot arm.

If a Generalized Inverse of a non-square $\boldsymbol{J}(\boldsymbol{q})$ is used (eq. (2.10)), any positive definite weighting matrix can be chosen. Without a meaningful choice, the resulting pseudo-inverse of $\boldsymbol{J}(\boldsymbol{q})$ incorporates no physical meaning (Azad and Babič, 2019).

Fig. 2.3 shows different manipulability ellipsoids for the same robot with the same body-fixed coordinate frame Ψ_{Task}. One ellipsoid has a randomly chosen weighting matrix $\boldsymbol{W}_{\text{rand}}$, with values given in appendix A.2. This illustrates how the obtained results critically depend on the choice of joint coordinates and the weighting matrix $\boldsymbol{W}$. In theory, one could even modify the elements of the weighting matrix to obtain any desired result (fig. 2.3). It is therefore necessary to mention the specific choices made when using $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ for manipulability analysis. Without that information, the result of the analysis cannot be compared between different users.

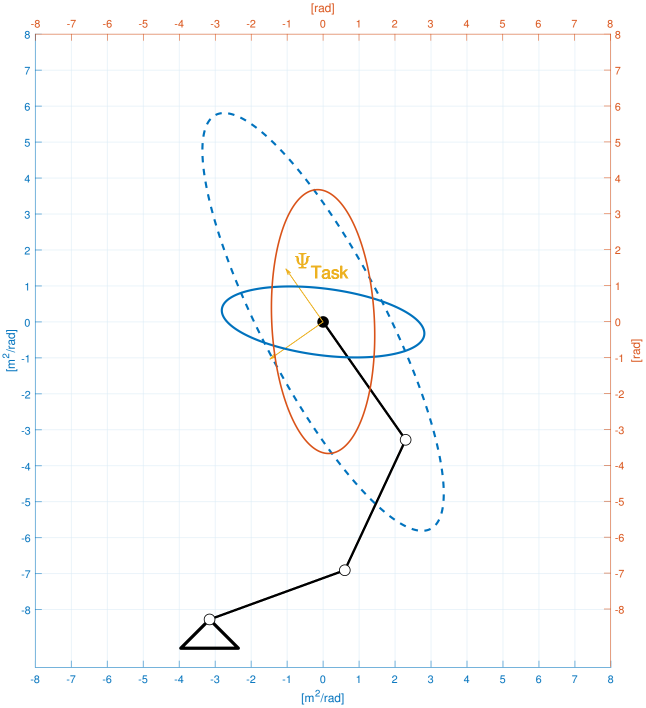


Figure 2.3: Velocity Manipulability Ellipsoids: blue ellipsoids for joint coordinate set $(\theta_1, \theta_2, \theta_3)$ and red ellipsoid for joint coordinate set (p_1, p_2, p_3). The ellipsoids drawn solid have $\boldsymbol{W} = \boldsymbol{I}$. The blue ellipsoid with dotted line has $\boldsymbol{W} = \boldsymbol{W}_{\text{rand}}$. $\boldsymbol{W}_{\text{rand}}$ and the values for the axes lengths and axes directions can be seen in appendix A.2.

2.2.3 Tensors for robotic dexterity analysis

Tensors are the basic objects of a coordinate invariant formulation of linear algebra and are represented by arrays and matrices. They can furthermore be made position dependent on a manifold and are then called "tensor fields." The dual objects of vectors are called "co-vectors." Often, no separation between vectors and co-vectors is made in robotics. Therefore, important information about the underlying elements gets lost. Tensor analysis separates these elements according to their transformation properties which make their physical nature explicit (appendix C.1). Using the rules of tensor contraction, the coordinate dependency of mathematical equations can be checked (appendix C.2). This knowledge will be used to analyze the derivation of $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ (sec. 2.2.1). Literature on tensors can be found in Boothby (1975), Dubrovin et al. (1984), and Frankel (2011).

The vector $\dot{\boldsymbol{q}}$ and its transpose representation $\dot{\boldsymbol{q}}^T$ in eq. (2.8) is a tensor of type $\binom{1}{0}$. To contract these two tensors to a scalar value, a quadratic form of type $\binom{0}{2}$ is needed (cf. appendix C.2). The mass matrix $\boldsymbol{M}(\boldsymbol{q})$ is such a tensor, which can be seen in eq. (2.14) for kinetic co-energy[4]:

$$\mathbb{R} \ni \mathscr{T} = \frac{1}{2}\dot{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{2.14}$$

This means that whatever joint coordinates are chosen, $\mathscr{T}$ will always have the same value because any change of coordinates for $\dot{\boldsymbol{q}}$ will imply a change of coordinates for $\boldsymbol{M}(\boldsymbol{q})$. The set of $\dot{\boldsymbol{q}}$ satisfying

$$\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = 1 \tag{2.15}$$

is an ellipsoid in the space of joint velocities at the configuration $\boldsymbol{q}$. It is an important first result that unlike eq. (2.8) with weighting matrix $\boldsymbol{W} = \boldsymbol{I}$, the ellipsoid in eq. (2.15) is relating the change of joint velocities to a change of the components of $\boldsymbol{M}(\boldsymbol{q})$ such that the scalar image of the quadratic form is coordinate invariant. Therfore, the ellipsoid can be computed, even if not all elements of $\dot{\boldsymbol{q}}$ have the same units.

After a specific choice for a weighting matrix in eq. (2.10), joint velocities $\dot{\boldsymbol{q}}$ can be determined which are inserted into eq. (2.15). This results in another workspace ellipsoid:

$$\boldsymbol{V}^T\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1}\boldsymbol{V} = 1. \tag{2.16}$$

This ellipsoid was called “Generalized Inertia Ellipsoid” in Asada (1983) and was recommended to be used for the design phase of the robot. Compared to the ellipsoid in eq. (2.12), this ellipsoid can be computed no matter which joint coordinates are used to represent the kinematic structure. Moreover, this ellipsoid incorporates dynamic characteristics of the robot. Nevertheless, this ellipsoid is only well-defined for fullrank $\boldsymbol{J}(\boldsymbol{q})$. The matrix elements of

$$\boldsymbol{\Lambda} = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1} \tag{2.17}$$

grow without bound near singular configurations.

2.3 Dexterity analysis for dynamic robot tasks

In the sequel of this chapter, the Lie group structure of $SE(3)$ will be utilized and the Geometrical Jacobian Matrix $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$ (eq. (2.7)) will be used. To make the notation light it will be defined: $\boldsymbol{J}(\boldsymbol{q}) \coloneqq \boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$.

[4] Kinetic co-energy is a quadratic form in flow (generalized velocity) and kinetic energy is a quadratic form in effort (generalized momentum) (Paynter et al., 1961)

To talk about the length or the norm of a vector, an inner product with metric g has to be used (sec. 2.1.3). For robotic dexterity analysis, the question arises: What would be a suitable choice for g? In the literature, g is often chosen to be the identity matrix due to the scalar product structure of $\mathbb{R}^3$, with orthogonal bases $b_1 = (1,0,0)$, $b_2 = (0,1,0)$, and $b_3 = (0,0,1)$. However, the inner product structure on $\mathbb{R}^3$ cannot be similarly used for $se(3)$, since the orthogonality has dependencies on the chosen coordinate frame (Lipkin and Duffy, 1988; Murray et al., 1994).

For the joint space, the mass matrix $\boldsymbol{M}(\boldsymbol{q})$ is a $\binom{0}{2}$-tensor that creates a natural bijective relation between the vector space $T_{\boldsymbol{q}}Q$ and the co-vector space $T^\star_{\boldsymbol{q}}Q$. Since $\boldsymbol{M}(\boldsymbol{q})$ is positive-definite, it can always be inverted. $\boldsymbol{M}(\boldsymbol{q})^{-1}$ can be mapped[5] to the workspace (fig. 2.4):

$$\boldsymbol{\Lambda}^{-1} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T. \tag{2.18}$$

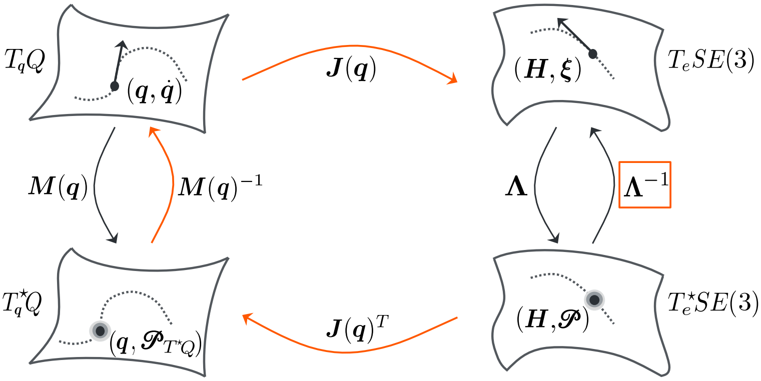

T_qQ
$(\boldsymbol{q},\dot{\boldsymbol{q}})$
$\boldsymbol{J}(\boldsymbol{q})$
$(\boldsymbol{H},\boldsymbol{\xi})$
$T_eSE(3)$
$\boldsymbol{M}(\boldsymbol{q})$
$\boldsymbol{M}(\boldsymbol{q})^{-1}$
$\boldsymbol{\Lambda}$
$\boxed{\boldsymbol{\Lambda}^{-1}}$
$T^\star_qQ$
$(\boldsymbol{q},\mathscr{P}_{T^\star Q})$
$\boldsymbol{J}(\boldsymbol{q})^T$
$(\boldsymbol{H},\mathscr{P})$
$T^\star_eSE(3)$

Figure 2.4: Graphical commutative diagram with tangent spaces $T_{\boldsymbol{q}}Q$ and $T_eSE(3)$ and co-tangent spaces $T^\star_{\boldsymbol{q}}Q$ and $T^\star_eSE(3)$ for the joint space (left) and workspace (right), respectively. The quadratic form $\boldsymbol{\Lambda}^{-1}$ maps (generalized) momentum $\mathscr{P}$ to twists $\boldsymbol{\xi}$. Starting from $T^\star_eSE(3)$, the composition of $\boldsymbol{\Lambda}^{-1}$ is represented by the orange arrows.

$\boldsymbol{\Lambda}^{-1}$ is a $\binom{2}{0}$-tensor. Since $\boldsymbol{\Lambda}^{-1}$ is induced by the use of the metric $\boldsymbol{M}(\boldsymbol{q})^{-1}$ in the joint space, it is invariant to the change of the inertial coordinate frame (left invariant).

With fig. 2.4, the reason for the dependency of $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ on joint coordinates becomes clear: Eq. (2.12) uses the identity matrix as metric and therefore can only be used if the chosen parameterization has the same unit for all robot joints.

$\boldsymbol{\Lambda}^{-1}$ maps (generalized) momenta $\mathscr{P} \in se^\star(3)$ to twists. In the further proceeding of this chapter maximum decoupling will be presumed and the coupling terms incorporated in $\boldsymbol{\Lambda}^{-1}$ will be ignored. In the work of Hogan (1984), $\boldsymbol{\Lambda^{-1}}$

[5] Also called “pulled back.”

is called "mobility end-point tensor." The matrix is associated to an admittance, which has "effort" as input and "flow" as output. It shows the admittance of the robot as seen from the task momentum. Also Khatib uses $\mathbf{\Lambda^{-1}}$ for his Operational Space Control Framework (Khatib, 1980, 1987). In Khatib (1995), $\mathbf{\Lambda^{-1}}$ was used to examine the inertial properties of the end-effector dynamics. Khatib and Sentis (2004) extended this work by analyzing the dynamic relation of a subordinate task and a main task (sec. 3.3).

For torque controlled robots, the dexterity of the robot is dependent on the inertial properties of the robot. Hence, the matrix $\mathbf{\Lambda}^{-1}$ is a good candidate for dynamic dexterity analysis. Compared to the quadratic form $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ (sec. 2.2.1), it brings three essential benefits: Firstly, through appropriate mapping between $T_eSE(3)$ and $T_e^{\star}SE(3)$, the calculation of $\mathbf{\Lambda}^{-1}$ doesn't need an inversion of $\boldsymbol{J}(\boldsymbol{q})$ (fig. 2.4). Secondly, $\mathbf{\Lambda}^{-1}$ is independent of the chosen joint coordinates, used to represent the kinematic structure of the manipulator (appendices A.2 and A.3). Thirdly, $\mathbf{\Lambda}^{-1}$ incorporates dynamic data and is therefore beneficial for the use of torque controlled robots. If it is desired to represent parts of $\mathbf{\Lambda}^{-1}$ in $\mathbb{R}^3$, $SE(3)$ has to be decomposed into translational and rotational subgroups and their respective sub-algebras have to be analyzed. The attributes of the translational and rotational sub-algebras can be seen in appendix B.

2.3.1 Screws: Geometric representation of Lie algebra structure

Any element of $se(3)$ has a geometric line in 3D-space associated with it. Hereby, the rotational part of the twist characterizes the direction of the line and the motion along the line. This geometric line will be called "a screw" and will be represented in Plücker coordinates with axis representation.

Definition 2.2 *A screw motion is defined as an instantaneous (pure) rotation about an axis together with an instantaneous (pure) translation along the same axis. Moreover, a pitch $h \in \mathbb{R}$ is defined as the ratio between translational motion $\boldsymbol{v}$ and rotational motion $\boldsymbol{w}$:*

$$\boldsymbol{\xi} = \underbrace{\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}}_{rotation} + \underbrace{h\begin{pmatrix} \boldsymbol{w} \\ 0 \end{pmatrix}}_{translation} .$$

Literature on "screw theory" can be found in Murray et al. (1994) and Stramigioli and Bruyninckx (2001a).

It can be seen in appendix B that $se(3)$ can be restricted to pure translations, i.e., twists with infinite pitch (eq. (B.4)). While a pure rotation (a twist with zero pitch) characterizes the line, a pure translation (twist with infinite pitch) is not bounded by a line.

Since the input of the map $\mathbf{\Lambda}^{-1}$ (eq. (2.18)) is a momentum $\mathscr{P}$, we are also interested in momentum screws. The fact that momentum is physically

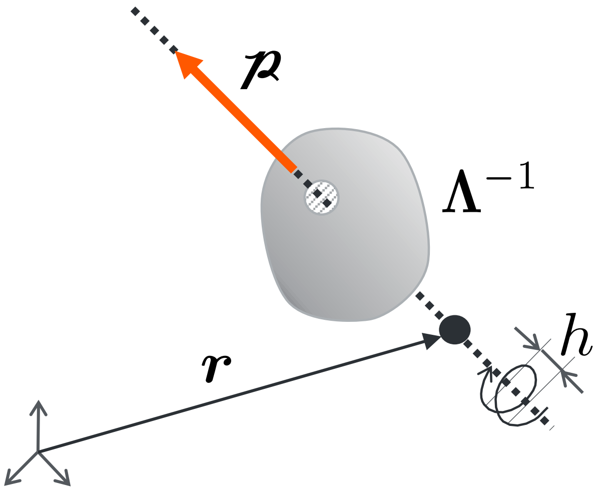


Figure 2.5: Representation of a momentum screw. The screw consists of linear momentum along the screw axis and rotational momentum about this axis. The pitch of the momentum screw is the ratio of angular momentum and linear momentum. A pure rotational momentum is a momentum screw with infinite pitch.

equal to a wrench and hence can be represented by a screw was shown in Stramigioli and Bruyninckx (2001b). The decomposition of momentum is dual to the decomposition of twists, with linear momentum $\boldsymbol{p} \in \mathbb{R}^3$ and rotational momentum $(\boldsymbol{r} \times \boldsymbol{p}) \in \mathbb{R}^3$:

$$\mathscr{P} = \underbrace{\begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{r} \times \boldsymbol{p} \end{pmatrix}}_{\text{linear momentum}} + \underbrace{h \begin{pmatrix} 0 \\ \boldsymbol{p} \end{pmatrix}}_{\text{rotational momentum}} . \tag{2.19}$$

In fig. 2.5, a momentum screw is visualized with linear momentum $\boldsymbol{p}$ along the screw axis. After choosing a point on the screw axis, the rotational momentum can be calculated by $\boldsymbol{r} \times \boldsymbol{p}$.

The knowledge gained from the Lie algebra structure (appendix B) and the association of $se^\star(3)$ with momentum screws will be used for dexterity analysis in the next sub-chapter.

2.3.2 Dexterity analysis

For dexterity analysis, often eigenvalue decomposition on a quadratic form is performed (tab. 2.1). Based on the eigenvalues and the direction of the corresponding eigenvectors, the robot dexterity can be represented as a sphere or an ellipsoid (fig. 2.3).

In tensor language, the commonly used “eigenvalue equation” (Lang, 1987) can be written as:

$$T^{a}{}_{m}\, u_a = \lambda\, u_m, \tag{2.20}$$

where $T^a_{\ m}$ is a linear operator with corresponding eigenvector u_a. The scalar λ is called the "eigenvalue" associated with u_m. It can be seen that this equality only holds for tensor $T^a_{\ m}$ of type $\begin{pmatrix}1\\1\end{pmatrix}$. To make eq. (2.20) work for a tensor T^{ka} of type $\begin{pmatrix}2\\0\end{pmatrix}$, a quadratic form can be used to lower one index of T^{ka} and obtain a linear operator (appendix C.2). A common choice is to use a (pseudo-)metric g_{mk}, which is defined on the same vector space as the tensor T^{ka}. Eq. (2.20) can therefore be rewritten:

$$g_{mk}\ T^{ka}\ u_a = \lambda\ u_m. \tag{2.21}$$

This is an important finding since the eigenvalues of a quadratic form—with reference to a (pseudo-)metric—are generally speaking the eigenvalues of a linear operator which arises through tensor contraction. The tensor contraction is well defined since the quadratic form and the (pseudo-)metric are defined on the same vector space.

For the analysis of the translational and rotational sub-algebras of $se(3)$, the bi-invariant quadratic form $\mathscr{M}$ (eq. (2.5)) will be used and specific scale choices will be made to apply tensor contraction on the $\begin{pmatrix}2\\0\end{pmatrix}$-tensor $\mathbf{\Lambda}^{-1}$.

Dexterity analysis for rotational tasks

For the analysis of rotational tasks, a first choice has to be made for the scale in eq. (2.5): $\beta = 0, \gamma = 1$. For $\gamma = 1$, $\mathscr{M}_K$ is called "Standard bi-invariant metric on SO(3)" in Park et al. (1993), Park and Brockett (1994), Park (1995), and is equal to the Euclidean metric. $\mathscr{M}_K$ is a $\begin{pmatrix}0\\2\end{pmatrix}$-tensor that can be used for tensor contraction on one index of $\mathbf{\Lambda}^{-1}$:

$$\left(\mathbf{\Lambda}_\epsilon^{-1}\right)^k_{\ i} = \left(\mathbf{\Lambda}^{-1}\right)^{kj}\ \left(\mathscr{M}_K\right)_{ij}. \tag{2.22}$$

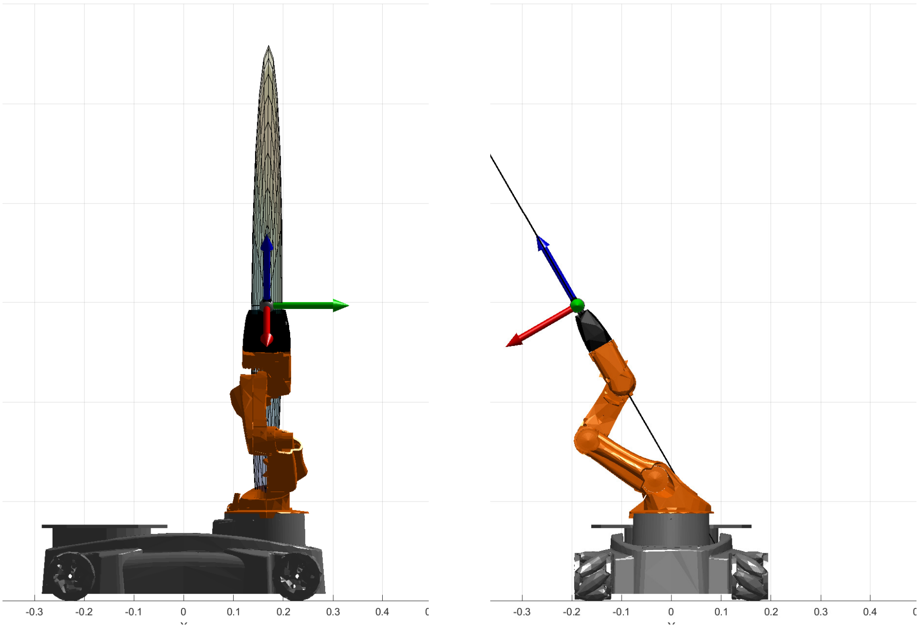
The action of $\mathscr{M}_K$ can directly be seen: After tensor contraction, the block matrix $\mathbf{\Lambda}^{-1}$ only contains one non-zero (3 × 3)-block, which is denoted $\mathbf{\Lambda}^{-1}_{\epsilon,\mathrm{3D}} \in \mathbb{R}^{3\times 3}$.

For a given rotational momentum $\mathbb{R}^3 \ni \boldsymbol{m} = (\boldsymbol{r} \times \boldsymbol{p})$ (momentum screw with infinite pitch), the angular velocity $\boldsymbol{w} \in \mathbb{R}^3$ of the end-effector can be calculated by:

$$\boldsymbol{w} = \mathbf{\Lambda}^{-1}_{\epsilon,\mathrm{3D}}\ \boldsymbol{m}. \tag{2.23}$$

As can be seen in fig. 2.5, a rotational momentum is not bounded by a line. To calculate a rotational momentum, a choice for a point $\boldsymbol{r} \in \mathbb{R}^3$ is needed.

Since $\mathbf{\Lambda}^{-1}_{\epsilon,\mathrm{3D}}$ is a $\begin{pmatrix}1\\1\end{pmatrix}$-tensor, the dependency on the choice for the body-fixed coordinate frame has been removed (appendix A.4). Once considering $\mathscr{M}_K$ as reference, eigenvalue decomposition on $\mathbf{\Lambda}^{-1}_{\epsilon,\mathrm{3D}}$ can be applied and can be represented as an ellipsoid (fig. 2.6). The ellipsoid is determined by the lengths and directions of the semi-axes. The axes point along the eigenvectors of $\mathbf{\Lambda}^{-1}_{\epsilon,\mathrm{3D}}$. The eigenvalues



(a) Ellipsoid in xz-view

(b) Ellipsoid in yz-view

Figure 2.6: *Rotational Dynamic Dexterity Ellipsoid* for a 8-DOF robot, with two translational coordinates and one rotational coordinate for the robot base and five rotational coordinates for the robot arm. The ellipsoid shows how well the robot can generate rotational motion in respective task coordinate indicated by the drawn task coordinate frame. About the blue coordinate-axis, high rotational motions can be produced, e.g., by turning the last joint. For rotational motion about the green coordinate-axis, high masses have to be moved. The red coordinate-axis shows the degenerating direction (cf. yz-view right). Moving the task coordinate on the end-effector body results in the same ellipsoid (cf. appendix A.4).

of $\mathbf{\Lambda}_{\epsilon,3\mathrm{D}}^{-1}$ determine the semi-axes length of the ellipsoid. The ellipsoid in fig. 2.6 shows how well a robot can transform rotational momentum to rotational motion. If $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{H}}$ is used for the calculation of $\mathbf{\Lambda}_{\epsilon,3\mathrm{D}}^{-1}$, the ellipsoid remains the same even if the coordinate frame on the robot body changes (appendix A.4).

Dexterity analysis for translational tasks

In the first step of the analysis of translational tasks, $\beta = 1, \gamma = 0$ (eq. (2.5)) and the Hyperbolic form $\mathscr{M}_H$ is used for tensor contraction:

$$\left(\bar{\mathbf{\Lambda}}^{-1}\right)^{k}_{\ i} = \left(\mathbf{\Lambda}^{-1}\right)^{kj} \left(\mathscr{M}_H\right)_{ij}. \tag{2.24}$$

Now, $\bar{\boldsymbol{\Lambda}}^{-1}$ is an anti-symmetric matrix. In a second step, $\mathscr{M}_K$ $(\beta = 0, \gamma = 1)$ can be used and tensor contraction on $\bar{\boldsymbol{\Lambda}}^{-1}$ can be applied:

$$\left(\boldsymbol{\Lambda}_{\mathrm{p}}^{-1}\right)_{il} = \left(\bar{\boldsymbol{\Lambda}}^{-1}\right)^{k}_{\ i} \left(\mathscr{M}_K\right)_{kl}. \tag{2.25}$$

With this contraction, the pure translational part of $\boldsymbol{\Lambda}^{-1}$ is extracted. Hence, the matrix $\boldsymbol{\Lambda}_{\mathrm{p}}^{-1}$ only contains one non-zero (3×3)-block, which is denoted $\boldsymbol{\Lambda}_{\mathrm{p,3D}}^{-1} \in \mathbb{R}^{3\times3}$. Compared to $\boldsymbol{\Lambda}_{\epsilon,\mathrm{3D}}^{-1}$, the tensor $\boldsymbol{\Lambda}_{\mathrm{p,3D}}^{-1}$ is of type $\begin{pmatrix}0\\2\end{pmatrix}$.

For a given linear momentum $\boldsymbol{\mathscr{p}} \in \mathbb{R}^3$ (momentum screw with zero pitch), the linear velocity $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ of the end-effector can be calculated by:

$$\dot{\boldsymbol{p}} = \boldsymbol{\Lambda}_{\mathrm{p,3D}}^{-1}\ \boldsymbol{\mathscr{p}}. \tag{2.26}$$

As can be seen in fig. 2.5, $\boldsymbol{\mathscr{p}}$ specifies the direction of the momentum screw axis. The direction and magnitude of $\boldsymbol{\mathscr{p}}$ is equal even though the co-vector is shifted along the line.

The matrix $\boldsymbol{\Lambda}_{\mathrm{p,3D}}^{-1}$ can be analyzed along specific directions. With given task direction $\boldsymbol{u} \in \mathbb{R}^3$, the scalar mass $m_u \in \mathbb{R}$ can be calculated that counteracts a linear momentum (Wassink and Stramigioli, 2007):

$$m_u = \left(\boldsymbol{u}^T \boldsymbol{\Lambda}_{\mathrm{p,3D}}^{-1} \boldsymbol{u}\right)^{-1}. \tag{2.27}$$

In the literature, the scalar m_u is often used for safety considerations in pHRI, as will be shown in chapter 4.

2.4 Conclusion

The first aim of this chapter is to make the reader aware of the influence of coordinates in robotic dexterity analysis. The context is not new in differential geometry, but often disregarded in robotics. First, an analysis of the dexterity measure $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ is performed. For the design process of a robot, $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ can be used to examine the robot's ability to transform joint velocities to task velocities. It therefore incorporates kinematic characteristics of the robot. The limitations of this measure are shown: $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ is not comparable for users with different coordinate choices. Moreover, for kinematically redundant robots, it depends on a choice for a cost function and a weighting matrix for the Jacobian pseudo-inverse. $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T$ cannot be used for robots with mixed joint coordinates (e.g., mobile robots).

The second aim of this chapter is to show how these limitations can be removed by applying tensor analysis. The derivation of $\boldsymbol{\Lambda}^{-1}$ is shown, which is invariant of joint coordinates. The key features of this matrix are:

1. $\boldsymbol{\Lambda}^{-1}$ shows a physically consistent mapping between the two robotic spaces. The choice of $\boldsymbol{M}(\boldsymbol{q})^{-1}$ as a metric in the Q-space enables a "natural scaling" of translational and rotational motion.

2. The matrix assigns a physical relationship to a robotic task: It shows how well a robot can resolve (task) momentum and therefore incorporates a dynamic characteristic.
3. The values of the matrix elements in $\mathbf{\Lambda}^{-1}$ are unique, no matter what set of joint coordinates are chosen to describe the kinematic structure.
4. The proposed dexterity measure can also be applied to kinematically redundant robots. Hereby, the values of the matrix elements in $\mathbf{\Lambda}^{-1}$ are unique because no inversion of the Jacobian matrix is needed.
5. An analysis of $\mathbf{\Lambda}^{-1}$ can be applied for robots with mixed joint parameters, e.g., mobile robots, without separating base and robot arm.
6. $\mathbf{\Lambda}^{-1}$ can be used as basis for dexterity analysis. The analysis of $\mathbf{\Lambda}^{-1}$ has a dependency on task coordinates.

The manifold $SE(3)$ incorporates no bi-invariant metric. Therefore, $se(3)$ had to be decomposed and the linear and rotational sub-algebras had to be analyzed separately. The structure of both sub-algebras were shown and their geometric properties were represented by screws. The main findings are:

1. By choosing the Euclidean metric as reference, it is possible to extract the rotational part of $\mathbf{\Lambda^{-1}}$. The values of the matrix elements in $\mathbf{\Lambda}_{\epsilon}^{-1}$ are unique, no matter what joint coordinates are taken and wherever the task coordinate is placed.
2. $\mathbf{\Lambda}_{\epsilon}^{-1}$ describes the dynamic ability of the robot to perform a rotational task: It shows how well the robot can transform rotational momentum to rotational motion.
3. $\mathbf{\Lambda}_{\epsilon}^{-1}$ can be used for dexterity analysis of purely rotational tasks. Once considering the Euclidean metric as reference, the Eigenvalues and Eigenvectors of $\mathbf{\Lambda}_{\epsilon}^{-1}$ can be calculated to plot a coordinate invariant Rotational Dynamic Dexterity Ellipsoid.
4. On $SE(3)$, no coordinate invariant way exists to extract the translational part of $\mathbf{\Lambda^{-1}}$.
5. Linear momentum is independent of the position on the screw axis. $\mathbf{\Lambda}_{\mathrm{p}}^{-1}$ can be analyzed by assigning specific directions. The scalar mass counteracting the linear momentum is often used for safety analysis.

2.4.1 Limitations

One practical limitation of the proposed dexterity measure is its use of the inverse mass matrix $\boldsymbol{M}(\boldsymbol{q})^{-1}$ as metric. For users without access to the robot dynamics, extensive work has to be done to identify the data. However, multiple identification methods exist (cf. overview provided in Siciliano and Khatib (2008)).

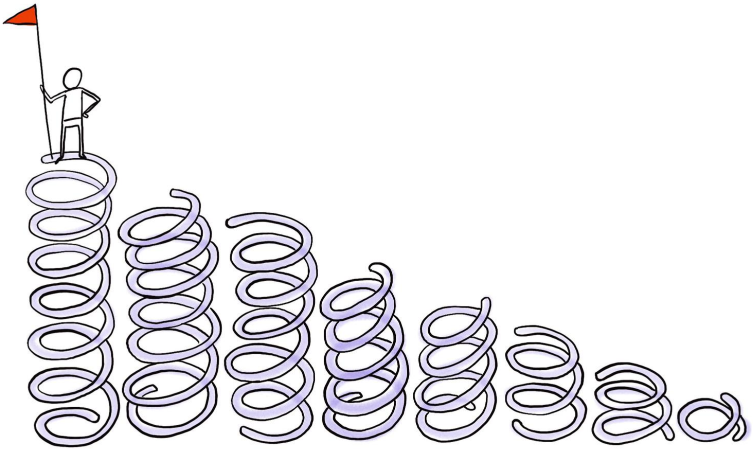
2.4.2 Towards coordinate-invariant dexterity analysis

Most industrial robots have six degrees of freedom and consist of purely revolute joints. Therefore, the choice for an appropriate dexterity measure is easy since (far away from singularities) the mapping between workspace and joint space is bijective. This means that the Jacobian inverse is unambiguously defined and hence no cost function and weighting matrix has to be chosen. To enlarge the feasible workspace, kinematically redundant robots can be used. Often, these robots don't posses unitary joint types. An example is a mobile robot with robot arm and platform. The dexterity measure should treat such a robot as one kinematic structure and the analysis should not be separated for robot arm and platform.

Traditional robot applications have stationary workspaces. Since the robot position and the workpiece have a fixed positions in space, the robot process is programmed by using coordinate frames. If the position and orientation of the coordinate frames remains constant, the dexterity of the robot might be described as a function of the stationary or body-fixed coordinate frame and the fixed robot posture. An example is screw fastening, where the dexterity along the screw axis could be described. Collaborative robots, however, interact with the environment (e.g., the human). Now the workspace is not fixed anymore, since it is influenced by the environment. As a consequence, the coordinate frames and the robot trajectory cannot remain constant and have to be adapted during task operation. An example is a humanoid robot with a robotic hand that picks the workpiece differently each cycle. For such an example, the dexterity measure should be applicable, even though the coordinate frames change.

CHAPTER 3

Shaping impedances to comply with constrained task dynamics



Lachner J, Allmendinger F, Stramigioli S and Hogan N (2022) Shaping impedances to comply with constrained task dynamics. *IEEE Transactions on Robotics* doi: 10.1109/TRO.2022.3153949

3.1 Introduction

For traditional industrial robots, usually one process-specific task is assigned. Since these robots have high inertia and operate at high speed, the robot cell is surrounded by a safety fence. Before the robot is installed, extensive planning has to be performed via CAD tools and simulations. The goal of the planning phase is to ensure that the robot trajectories are within the robot workspace and are not restrained by physical obstacles. Once the planning phase is finished, the set-up begins and the robot motion is programmed. Unfortunately, kinematic singularities are encountered often during programming. It is the task of the robot programmer to determine suitable singularity-free trajectories.

In the meanwhile, a considerable number of collaborative robots exist (International Federation of Robotics, 2019a). Since these robots operate next to a human co-worker, additional features are required, specifically stability and safe behavior in contact (Stramigioli, 2001). Often, these robots are mobile and possess a high number of degrees of freedom. This makes them more dexterous and hence they are able to perform multiple simultaneous tasks. Next to kinematic singularities, conflicts between multiple simultaneous tasks impede robot programming. Therefore, the controller has to be reactive and the control parameters have to be adapted autonomously during task execution.

Impedance control, originally proposed by Hogan (1984), is a well-established control method for robot applications that involve physical interaction. The control method already implements many of the required features for collaborative robots: 1) An impedance-controlled robot can be used in contact and out of contact with the environment (Hogan, 1985a); 2) The control method has beneficial stability properties and hence can be used for interaction with most environments (Hogan, 1988); 3) Multiple impedances can be superimposed, even if they are non-linear (Hogan, 1985b); 4) To control kinematically redundant robots, no inverse kinematics are needed and hence the robot can go in and out of singularities without further singularity handling by the controller (Mussa-Ivaldi and Hogan, 1991).

Nevertheless, impedance control also has its limitations. If multiple impedances are superimposed, they usually influence each other. This has the consequence that the desired end-effector behavior might suffer (Hermus et al., 2022). Moreover, for a given task, advanced knowledge is needed to select feasible impedance parameters. For non-expert users, the selection and composition of appropriate impedance parameters is not trivial (Hogan, 1985a; Buchli et al., 2011; Martin-Martin et al., 2019).

Chapter 2 addressed the foundations of robot dexterity and showed how kinematic singularities can be detected. This chapter will show how a physical threshold can be set up to help the robot programmer detect a kinematic singularity before it occurs. The methodology will be extended to detect conflicts between multiple simultaneous tasks.

3.1.1 Task hierarchy

For kinematically redundant robots, infinitely many sets of joint velocities exist that create a desired task velocity. It is possible to search for the joint velocity set that also fullfills additional tasks. To find a unique joint velocity set, an optimization problem may be formulated. Most commonly, nullspace projection methods are used that resolve the redundancy via the Jacobian pseudo-inverse (eqs. (2.9) - (2.11)) and the choice for a "nullspace projection matrix" (Khatib, 1987; Nakamura et al., 1987; Siciliano and Slotine, 1991). Another common approach uses quadratic programming. Dedicated solvers are applied to minimize an objective function (Faverjon and Tournassoud, 1987; Kanoun et al., 2011). For all those control approaches, a task hierarchy is established with a main task as highest priority and subordinate tasks with lower priority. The subordinate task of lowest priority usually controls the motion in the nullspace of $\boldsymbol{J}(\boldsymbol{q})$.

The priorities are usually assigned by the programmer of the robot application. Hereby, implementations with "soft" and "strict" hierarchies exist (Dietrich et al., 2015). Especially the implementation of strict hierarchies is computationally expensive (Kanoun et al., 2011; Escande et al., 2014).

In this chapter, task priorities are labeled with a superscript. The superscripts are numbered from 1 (main task) to 2 (subordinate task with second highest priority) to $i \in \mathbb{R}$ (task with lowest priority).

3.1.2 Task conflicts

Depending on the task (i.e., task dimension, desired kinematic/dynamic behavior, goal position) and the given kinematic structure (i.e., degrees of freedom, kinematic/dynamic ability, available workspace), the robot might not be able to perform multiple tasks simultaneously. If a task hierarchy is established, tasks with lower priority will be sacrificed during a task conflict. For many applications, however, it is crucial that the robot indicates that not all tasks are mutually compatible. There are only few methods in the literature that can be used during robot operation to predict a task conflict (Schettino et al., 2021).

The main contribution of this chapter is an extension for a Cartesian impedance controller that shapes impedances in order to handle conflicts between multiple tasks with different priorities. The optimization is executed in parallel to the control structure. This has computational advantages and is especially useful for robots with many DOF. The controller ensures the stability of the robot by monitoring the transmitted power through what will be introduced as virtual springs and the supplied power from the controller to the robot. In case excessive power would be supplied to the robot, the impedance optimization is adapted to guarantee stability. This makes the control method suitable for robots that involve pHRI. The method is validated on a real robot and compared to well-established control methods. Table 3.1 shows an overview of the symbols, superscripts, and supscripts used throughout this chapter. The results of this chapter are partly reported in Lachner et al. (2022).

Table 3.1: Overview of the used symbols, superscripts, and subscripts.

	Symbols
$'$	Impedance shaping
$\hat{}$	Inertia shaping (Khatib, 1987)
$\tilde{}$	Screw-symmetric matrix form
	Superscripts, indicating task priority
1	Main task
$2-i$	Subordinate tasks
c	Task consistency
	Subscripts, indicating task attribute
p	Translational task
ϵ	Rotational task
	Control torques of experiments
τ^{imp}	Impedance superposition
τ^{np}	Nullspace projection
τ'	Impedance shaping

3.2 Cartesian impedance control based on potential energy functions

The equation of motion for a robot with n joints can be represented by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}^{\mathrm{imp}} + \boldsymbol{\tau}^{\mathrm{ext}}, \tag{3.1}$$

with $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ being the symmetric and positive definitive inertia matrix and $\boldsymbol{g}(\boldsymbol{q}) = \partial \mathscr{U}_{\mathrm{g}}(\boldsymbol{q})/\partial \boldsymbol{q} \in \mathbb{R}^n$ being the force resulting from the gravitational potential energy $\mathscr{U}_{\mathrm{g}}(\boldsymbol{q})$. The centrifugal/Coriolis matrix $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ is defined via the Christoffel symbols (Murray et al., 1994) such that $\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T$ holds, which implies that $\dot{\boldsymbol{q}}^T(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}))\dot{\boldsymbol{q}} = 0$. The control torque of the robot is $\boldsymbol{\tau}^{\mathrm{imp}} \in \mathbb{R}^n$. $\boldsymbol{\tau}^{\mathrm{ext}} \in \mathbb{R}^n$ are the total resultant effects of all external forces as torques acting on the joints. With the definition of $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, the robot is passive with respect to torque input $\boldsymbol{\tau}^{\mathrm{imp}} + \boldsymbol{\tau}^{\mathrm{ext}}$ and motion output $\dot{\boldsymbol{q}}$. The passivity property of the robotic system will be treated in more detail later in the chapter.

In eq. (2.7), a stationary inertial coordinate frame "0" was chosen to calculate the Geometrical Jacobian Matrix. Instead of an inertial coordinate frame, a body-fixed coordinate frame "b" can also be chosen. This yields the "Geometrical Body Jacobian Matrix" (Murray et al., 1994) $\boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}$ that maps joint velocities $\dot{\boldsymbol{q}}$ to body twists in vector form $\boldsymbol{\xi}^{\mathrm{b}} \in \mathbb{R}^6$:

$$\boldsymbol{\xi}^{\mathrm{b}} = \boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}\ \dot{\boldsymbol{q}}. \tag{3.2}$$

The vector $\boldsymbol{\xi}^{\mathrm{b}}$ is expressed in the body-fixed coordinate frame with respect to the inertial coordinate frame. The columns of $\boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}$ are the twist axes of the robot joints with respect to the body-fixed coordinate frame in the configuration $\boldsymbol{q}$. $\boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}$ is related to $\boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}$ (eq. (2.7)) by:

$$\boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}} = \boldsymbol{Ad}^{-1}_{({}^{0}\boldsymbol{H}_{\mathrm{b}})}\, \boldsymbol{J}(\boldsymbol{q})_{\mathrm{G}}, \tag{3.3}$$

where $\boldsymbol{Ad}^{-1}_{({}^{0}\boldsymbol{H}_{\mathrm{b}})} = \boldsymbol{Ad}_{({}^{\mathrm{b}}\boldsymbol{H}_{0})} \in \mathbb{R}^{6\times 6}$ is the inverse of the “Adjoint Matrix” (Murray et al., 1994):

$$\boldsymbol{Ad}_{({}^{0}H_{\mathrm{b}})} = \begin{pmatrix} {}^{0}\boldsymbol{R}_{\mathrm{b}} & {}^{0}\tilde{\boldsymbol{p}}_{\mathrm{b}}\,{}^{0}\boldsymbol{R}_{\mathrm{b}} \\ 0 & {}^{0}\boldsymbol{R}_{\mathrm{b}} \end{pmatrix}. \tag{3.4}$$

${}^{0}\boldsymbol{R}_{\mathrm{b}} \in SO(3)$ and ${}^{0}\tilde{\boldsymbol{p}}_{\mathrm{b}} \in \mathbb{R}^{3}$ are the rotation matrix and the position (in skew-symmetric matrix form) between the body-fixed coordinate frame and the inertial coordinate frame.

In the rest of the chapter, all entities will be expressed in body-fixed coordinates. Hence, to make the notation light it will be defined: $\boldsymbol{J}(\boldsymbol{q}) := \boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}$ and $\boldsymbol{\xi} := \boldsymbol{\xi}^{\mathrm{b}}$.

For a Cartesian impedance controller, the task is represented by a virtual spring that pulls the robot body (e.g., end-effector body) from its current pose “b” towards a desired equilibrium pose “e.” The homogeneous transformation between “b” and “e” can be represented by the matrix ${}^{\mathrm{b}}\boldsymbol{H}_{\mathrm{e}} \in SE(3)$. ${}^{\mathrm{b}}\boldsymbol{H}_{\mathrm{e}}$ incorporates the displacement $\Delta\boldsymbol{p} = ({}^{0}\boldsymbol{p}_{\mathrm{e}} -{}^{0}\, \boldsymbol{p}_{\mathrm{b}}) \in \mathbb{R}^{3}$ and the rotation ${}^{\mathrm{b}}\boldsymbol{R}_{\mathrm{e}} \in SO(3)$:

$${}^{\mathrm{b}}\boldsymbol{H}_{\mathrm{e}} = \begin{pmatrix} {}^{\mathrm{b}}\boldsymbol{R}_{\mathrm{e}} & \Delta\boldsymbol{p} \\ 0 & 1 \end{pmatrix}. \tag{3.5}$$

Here, the equilibrium position is denoted with ${}^{0}\boldsymbol{p}_{\mathrm{e}}$ and the selected position on the robot body is denoted with ${}^{0}\boldsymbol{p}_{\mathrm{b}}$.

As discussed in sec. 2.1.1, global and local representations of rotations exist. ${}^{\mathrm{b}}\boldsymbol{R}_{\mathrm{e}}$ is a unique and global representation of the rotation between “b” and “e.” Another global representation of rotations are unit-length quaternions. The rotation matrix ${}^{\mathrm{b}}\boldsymbol{R}_{\mathrm{e}}$ can be transformed to unit-length quaternions by using:

$$\tilde{\boldsymbol{\epsilon}}\ \eta = \log({}^{\mathrm{b}}\boldsymbol{R}_{\mathrm{e}}), \tag{3.6}$$

with scalar rotation angle $\eta \in \mathbb{R}$ and unit-axis of rotation $\tilde{\boldsymbol{\epsilon}} = -\tilde{\boldsymbol{\epsilon}}^{T} \in so(3)$, in skew-symmetric matrix form. The unit-length quaternion ${}^{\mathrm{b}}\boldsymbol{Q}_{\mathrm{e}}$ can then be calculated by:

$${}^{\mathrm{b}}\boldsymbol{Q}_{\mathrm{e}} = \cos(\frac{\eta}{2}) - \boldsymbol{\epsilon}\ \sin(\frac{\eta}{2}). \tag{3.7}$$

For the experiments of this thesis (sec. 3.5 and sec. 4.3.3), the algorithm to compute unit-length quaternions was based on Allmendinger (2015).

Due to the virtual spring, the robot is exposed to a potential energy function $\mathscr{U} : SE(3) \to \mathbb{R}$ where $\mathscr{U}$ has a translational and rotational part coupled together. Commonly, the translational part of $\mathscr{U}$ is a function of the displacement $\Delta\boldsymbol{p}$ and the translational stiffness $\boldsymbol{K}_{\mathrm{p}} \in \mathbb{R}^{3\times 3}$. In the experiments of this chapter (sec. 3.5), $\boldsymbol{K}_{\mathrm{p}}$ is chosen to be a diagonal matrix. The rotational part of $\mathscr{U}$ can be expressed as a function of the unit-axis $\boldsymbol{\epsilon}$ and the rotational stiffness $\boldsymbol{K}_\epsilon \in \mathbb{R}^{3\times3}$. Also $\boldsymbol{K}_\epsilon$ is chosen to be a diagonal matrix. Both, $\boldsymbol{K}_\text{p}$ and $\boldsymbol{K}_\epsilon$ are expressed in the body-fixed coordinate frame. The total potential $\mathscr{U} = \mathscr{U}_\text{p} + \mathscr{U}_\epsilon$, with respect to the body-fixed coordinate frame, is chosen to be:

$$\begin{aligned}\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon}) = &\underbrace{\frac{1}{2}\Delta\boldsymbol{p}^T\ \boldsymbol{K}_p\ \Delta\boldsymbol{p}}_{\mathscr{U}_\text{p}} + \\ &+ \underbrace{2\ \boldsymbol{\epsilon}^T\ \boldsymbol{K}_\epsilon\ \boldsymbol{\epsilon}}_{\mathscr{U}_\epsilon}.\end{aligned} \tag{3.8}$$

To ensure asymptotic convergence towards the equilibrium pose, a virtual damper is introduced that acts on the body twist $\boldsymbol{\xi}$ and/or the robot velocity $\dot{\boldsymbol{q}}$. For now it will be assumed that the controller directly damps $\dot{\boldsymbol{q}}$ and a more advanced damping design method will be presented in sec. 3.4.3. The dissipation can be described by a Rayleigh function $\mathscr{R} : \mathbb{R}^n \to \mathbb{R}$. One simple choice for such a function is:

$$\mathscr{R}(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{B}_{\boldsymbol{q}}\ \dot{\boldsymbol{q}}, \tag{3.9}$$

with positive definite damping matrix $\boldsymbol{B}_{\boldsymbol{q}} \in \mathbb{R}^{n\times n}$.

The control wrench $\boldsymbol{F}^\text{imp} \in se^\star(3)$ that is supposed to minimize the potential energy function $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ can be derived by computing the differential of $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ (appendix D). In the further proceeding of this chapter, the centrifugal/Coriolis effects will be neglected[1]. Since the robot is exposed to gravity, the controller has to compensate the gravitational potential energy by creating an artificial potential field that counterbalances the physical potential energy $\mathscr{U}_\text{g}(\boldsymbol{q})$. The control torque $\boldsymbol{\tau}^\text{imp} \in \mathbb{R}^n$ can be calculated as follows:

$$\boldsymbol{\tau}^\text{imp} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{F}^\text{imp} - \frac{\partial\mathscr{R}(\dot{\boldsymbol{q}})}{\partial\dot{\boldsymbol{q}}} + \frac{\partial\mathscr{U}_\text{g}(\boldsymbol{q})}{\partial\boldsymbol{q}}. \tag{3.10}$$

For an impedance-controlled robot, multiple tasks can be assigned. Each task can be expressed as a virtual spring-damper-system that acts on a point on the robot structure. This point usually coincides with the center of a body-fixed coordinate frame, placed on one of the robot bodies. Care has to be taken if multiple spring-damper-systems act on different points on the robot body. For different points, the Body Jacobian Matrices are different. First, each set of task torques has to be calculated that balances the individual spring-damper-system. Afterwards, the individual task torques can be superimposed. The same procedure has to be done if multiple spring-damper-systems act on points, placed on different robot body parts.

Through the superposition, an impedance controller assigns the same priority to a main task and all other (subordinate) tasks. How much these tasks influence each other depends on the equilibrium pose, the magnitude of the impedances, and the DOF of the robot (Hermus et al., 2022).

[1] A power continuous solution to compensate the centrifugal/Coriolis effects via feedback compensation can be found in Wu (2016)

3.2.1 Passivity property for passive environments

A controlled robotic system consists of multiple subsystems, i.e., the controller, the robot, and the environment (fig. 3.1).

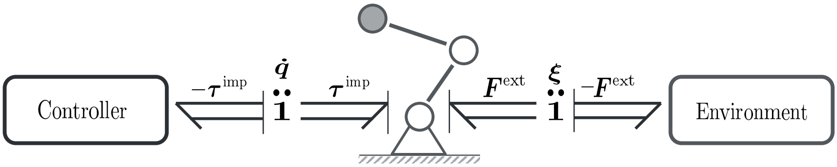

Controller
$-\boldsymbol{\tau}^{\text{imp}}$
$\dot{\boldsymbol{q}}$
$\mathbf{1}$
$\boldsymbol{\tau}^{\text{imp}}$
$\boldsymbol{F}^{\text{ext}}$
$\boldsymbol{\xi}$
$\mathbf{1}$
$-\boldsymbol{F}^{\text{ext}}$
Environment

Figure 3.1: Subsystems of a robotic system. The controller supplies power $(\boldsymbol{\tau}^{\text{imp}})^T\dot{\boldsymbol{q}}$ to the robot. The robot consists of motors and the kinematic structure. If the robot interacts with the environment, power $(\boldsymbol{F}^{\text{ext}})^T\boldsymbol{\xi}$ is exchanged with the environment. A negative sign indicates power flow from the controller to the robot and power flow from the environment to the robot, respectively. All elements with Bond Graph notation are shown in Appendix E.

The power supply from the controller to the robot can be described with two conjugate "power variables" (Stramigioli, 2001): $\boldsymbol{\tau}^{\text{imp}}$ (effort) and $\dot{\boldsymbol{q}}$ (flow). For robots that interact with the environment (e.g., during pHRI), energy is exchanged via the two conjugate power variables: $\boldsymbol{F}^{\text{ext}}$ (effort) and $\boldsymbol{\xi}$ (flow).

During an unstable behavior, the controller transfers excessive power to the robot that may not be needed for the desired control actions. If the stability of a robotic system must be guaranteed, it is necessary to supervise and control the energy flow from the controller to the robot in order to prevent excessive energy transfers (Stramigioli, 2001). For robots that interact with unknown environments, Stramigioli (2015) has formally proven the necessity of passivity-based control.

A robotic system is a dynamical system with input and output ports (fig. 3.1). Moreover, a robotic system is a dissipative system since a part of the electrical and mechanical free energy is dissipated by heat production through resistors and friction, respectively. It is shown in Willems (1972) that for dynamical dissipative systems, a finite storage function always exists which is the energy of the system. These dynamical dissipative systems can only store a part of the supplied energy and can only supply a part of what has been stored. For an impedance-controlled robot, the stored energy can be described by its energy function, called "storage function" $S \in \mathbb{R}$. Here, S is composed of the storage function of the robot $S^{\text{rob}} \in \mathbb{R}$ and the storage function of the controller $S^{\text{ctrl}} \in \mathbb{R}$. The controlled robotic system is passive with respect to the interaction port $(\boldsymbol{\xi}, \boldsymbol{F}^{\text{ext}})$ (fig. 3.1) if the robot and the controller are passive (Stramigioli, 2015) and are connected in a power continuous way.

A robot is a passive system since the storage function $S^{\text{rob}} = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \mathscr{U}_{\text{g}}(\boldsymbol{q})$ that describes the system has a lower bound due to physical reasons (Wyatt et al., 1981; van der Schaft, 2016).

In the concept of "control by interconnection" (Stramigioli, 2001; van der Schaft, 2016), a controller is not only seen as a simple signal processor but has a physical interpretation assigned to it. The controller itself can consist of multiple physical interpretable subsystems. An impedance controller consists of multiple virtual energetic subsystems that are interconnected with each other and preserve energy continuity. The subsystems can either supply energy, route energy among subsystems, store energy, or dissipate energy (Califano et al., 2021). The storage function of the controller S^{ctrl} can be described with energy functions and energy dissipation can be described with Rayleigh functions.

The storage function of the controller S^{ctrl} is the sum of the potential energy functions, i.e., $S^{\text{ctrl}} = \mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon}) - \mathscr{U}_g(\boldsymbol{q})$. Since only robots with rotational joints are considered in this chapter, $\mathscr{U}_g(\boldsymbol{q})$ has a lower bound.

Definition 3.1 *For a given storage function S^{ctrl}, a controller is said to be strictly output passive with respect to power port $(\dot{\boldsymbol{q}}, -\boldsymbol{\tau})$ if:*

$$\dot{S}^{ctrl} \leq -\boldsymbol{\tau}^T\ \dot{\boldsymbol{q}}.$$

The energy flow $\dot{S}^{\text{ctrl}}$ from the robot to the controller (fig. 3.1) can be calculated by time differentiation of S^{ctrl}:

$$\dot{S}^{\text{ctrl}} = (\frac{\partial\mathscr{U}_{\text{p}}}{\partial\Delta\boldsymbol{p}})^T\ \dot{\Delta\boldsymbol{p}} + (\frac{\partial\mathscr{U}_{\epsilon}}{\partial\boldsymbol{\epsilon}})^T\ \dot{\boldsymbol{\epsilon}} - (\frac{\partial\mathscr{U}_{\text{g}}(\boldsymbol{q})}{\partial\boldsymbol{q}})^T\ \dot{\boldsymbol{q}}. \tag{3.11}$$

Note that $\dot{\boldsymbol{\epsilon}} \in \mathbb{R}^3$ can be calculated with eq. (D.10). Considering that the control torques $\boldsymbol{\tau}^{\text{imp}}$ are supposed to minimize the potential energy functions (appendix D), the supplied power from the robot to the controller is:

$$\begin{aligned} -\boldsymbol{\tau}^{\text{imp}^T}\ \dot{\boldsymbol{q}} = {} & (\frac{\partial\mathscr{U}_{\text{p}}}{\partial\Delta\boldsymbol{p}})^T\ \dot{\Delta\boldsymbol{p}} + (\frac{\partial\mathscr{U}_{\epsilon}}{\partial\boldsymbol{\epsilon}})^T\ \dot{\boldsymbol{\epsilon}} \\ & - (\frac{\partial\mathscr{U}_{\text{g}}(\boldsymbol{q})}{\partial\boldsymbol{q}})^T\ \dot{\boldsymbol{q}} + (\frac{\partial\mathscr{R}(\dot{\boldsymbol{q}})}{\partial\dot{\boldsymbol{q}}})^T\ \dot{\boldsymbol{q}}. \end{aligned} \tag{3.12}$$

By comparing eq. (3.11) and eq. (3.12) with the passivity condition of def. 3.1, and considering the boundedness of $\mathscr{U}_g(\boldsymbol{q})$, it can be concluded that the controller is strictly output passive with respect to interaction port $(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}^{\text{imp}})$.

Bond Graphs, originally invented by Paynter et al. (1961), provide a graphical representation to analyze the interconnection and power flow between multiple subsystems. An advantageous feature of Bond Graphs is that they can be used independently of the system domain (e.g., mechanical, electrical or chemical). It should be noted that Bond Graphs have also additional advantages that will not be used in this thesis, e.g., they simplify the derivation of differential equations of a given system.

Fig. 3.2 shows the Bond Graph representation of a simple controlled robotic system. The potential energy function $\mathscr{U}_g(\boldsymbol{q})$ and the centrifugal/Coriolis terms $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ of the robot are defined in eq. (3.1). The controlled potential energy function $\mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})$ is defined in eq. (3.8) and the Rayleigh function $\mathscr{R}(\dot{\boldsymbol{q}})$ is defined in eq. (3.9). All elements of the Bond Graph model are explained in Appendix E. A more rigorous treatment of modelling and control of physical systems with Bond

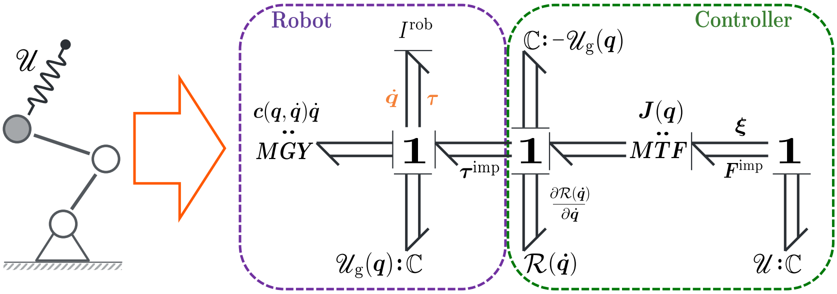

Robot
Controller
$\mathscr{U}$
I^{rob}
$\dot{\boldsymbol{q}}$
$\boldsymbol{\tau}$
$c(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$
$\ddot{MGY}$
1
$\boldsymbol{\tau}^{\mathrm{imp}}$
$\mathscr{U}_{\mathrm{g}}(\boldsymbol{q})$:ℂ
ℂ: $-\mathscr{U}_{\mathrm{g}}(\boldsymbol{q})$
1
$\frac{\partial \mathcal{R}(\dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}$
$\mathcal{R}(\dot{\boldsymbol{q}})$
$\boldsymbol{J}(\boldsymbol{q})$
$\ddot{MTF}$
$\boldsymbol{\xi}$
$\boldsymbol{F}^{\mathrm{imp}}$
1
$\mathscr{U}$:ℂ

Figure 3.2: Bond Graph model of a simple robot connected to potential energy due to a virtual spring. All elements with Bond Graph notation are shown in Appendix E.

Graphs can be found in Duindam et al. (2009) and Folkertsma and Stramigioli (2015).

The power flow in the Bond Graph model is represented with double-lined arrows, called "multi-bonds." The direction of the multi-bonds shows the positive direction of power flow. The origin of the flow variable $\dot{\boldsymbol{q}}$ (and equally the destination of the effort variable) is represented by a thin line attached to the multi-bond, which is called "causal stroke." For the simple Bond Graph example, it can be seen that the subsystems of the controller share the same flow-variable, indicated by a 1-junction, which represents the robot's joint motion $\dot{\boldsymbol{q}}$. Hence, the 1-junction sums the efforts of the individual subsystems (individual torques of eq. (3.10)). These efforts can be derived by the partial derivative of $\mathscr{U}_g(\boldsymbol{q})$, the partial derivative of $\mathscr{R}(\dot{\boldsymbol{q}})$, and by the differential of $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ (appendix D). In Bond graph notation, stored potential energy is represented by a ℂ-element. For the potential energy function $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$, the Body Jacobian Matrix $\boldsymbol{J}(\boldsymbol{q})$ (eq. (3.3)), together with its transpose $\boldsymbol{J}(\boldsymbol{q})^T$, establishes a power continuous transformation of power conjugate variables $(\boldsymbol{\xi},\boldsymbol{F}^{\mathrm{imp}})$ in the workspace to the power conjugate variables $(\dot{\boldsymbol{q}},\boldsymbol{\tau})$ in the joint space. In Bond Graph notation, a power continuous transformation that can be modulated is represented by a "modulated transformer element" (Duindam et al., 2009) $\boldsymbol{MTF}$. In this example, the power continuous transformation is modulated by the joint configuration $\boldsymbol{q}$.

With $\dot{\boldsymbol{q}}$ being a flow, the Bond Graph model shows the causality of energy flow between the subsystems of the controller and the robot. The robot, represented by the inertial element I^{rob}, starts moving since it is exposed to potential energy $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$. Once the robot moves, power is virtually flowing between the subsystems of the controller and energy is transferred via the power port $(\dot{\boldsymbol{q}},\boldsymbol{\tau}^{\mathrm{imp}})$. Once $\dot{\boldsymbol{q}} \neq 0$, a part of the stored potential energy is dissipated, represented by the Rayleigh function $\mathscr{R}(\dot{\boldsymbol{q}})$. This shows that for a given Bond Graph model of a controlled robotic system, the passivity analysis can be made by visual inspection.

3.2.2 Nullspace projection

To map joint torques to workspace wrenches, a solution for the Generalized Inverse of $\boldsymbol{J}(\boldsymbol{q})$ has to be found (eq. (2.10)). As shown in Khatib (1987), a solution for the Jacobian right-inverse $\bar{\boldsymbol{J}}(\boldsymbol{q}) \in \mathbb{R}^{n\times 6}$ that satisfies the system dynamics is:

$$\bar{\boldsymbol{J}}(\boldsymbol{q}) \coloneqq \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1}. \tag{3.13}$$

Outside singularities, the transpose of $\bar{\boldsymbol{J}}(\boldsymbol{q})$ can be used to determine workspace wrenches for a given set of joint torques.

All torques that balance workspace wrenches lie in the Row Space of $\bar{\boldsymbol{J}}(\boldsymbol{q})^T$. There are additional torques that can be controlled without producing undesired workspace wrenches (or accelerations). These torques lie in the Nullspace of $\bar{\boldsymbol{J}}(\boldsymbol{q})^T$.

In linear algebra, a projection is a linear map from a (co-)vector space onto itself. A projection matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is idempotent, which means $\boldsymbol{PP} = \boldsymbol{P}$ (Dubrovin et al., 1984). The projection matrix onto the Row Space of $\bar{\boldsymbol{J}}(\boldsymbol{q})^T$ is:

$$\boldsymbol{P} = \boldsymbol{J}(\boldsymbol{q})^T\ \bar{\boldsymbol{J}}(\boldsymbol{q})^T. \tag{3.14}$$

The complementary projection matrix $\boldsymbol{N} \in \mathbb{R}^{n\times n}$, that projects onto the Nullspace of $\bar{\boldsymbol{J}}(\boldsymbol{q})^T$, is:

$$\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{J}(\boldsymbol{q})^T\ \bar{\boldsymbol{J}}(\boldsymbol{q})^T. \tag{3.15}$$

By using nullspace projection, a hierarchy between multiple sets of task torques can be established. Here, two main implementation possibilities exist: the "successive" and the "augmented" methods (Dietrich et al., 2015). Since the successive method doesn't guarantee the decoupling of all tasks, the priority assignment is called "soft." Augmented methods, however, assign "fixed" task priorities by decoupling all task hierarchies (Flacco et al., 2012). In this work, successive projectors will be used, which assign a soft priority to each individual task torque. An overview and comparison of both nullspace projection methods can be found in Dietrich et al. (2015).

A main task (related to the Jacobian matrix $\boldsymbol{J}^1(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ and torque $\boldsymbol{\tau}^1 \in \mathbb{R}^n$) and a subordinate task (related to the Jacobian matrix $\boldsymbol{J}^2(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ and torque $\boldsymbol{\tau}^2 \in \mathbb{R}^n$) can be composed by using the nullspace projection matrix of eq. (3.15):

$$\boldsymbol{\tau}^{\mathrm{np}} = \boldsymbol{\tau}^1 + \boldsymbol{N}^1\ \boldsymbol{\tau}^2. \tag{3.16}$$

Here, $\boldsymbol{\tau}^{\mathrm{np}} \in \mathbb{R}^n$ is the command torque which ensures that subordinate task accelerations do not interfere with the main task. This projection method can be extended by exploiting the nullspace of $\bar{\boldsymbol{J}}^2(\boldsymbol{q})^T$ to implement further tasks with lower priority.

On the right hand side of fig. 3.3, an exemplary projection $\boldsymbol{\tau}^{\mathrm{np}} = \boldsymbol{N}\boldsymbol{\tau}^{\mathrm{imp}}$ is illustrated. It can be seen that the nullspace projection is *not* power continuous since it only acts on the effort part of the interaction port $(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}^{\mathrm{imp}})$. Due to the projection, the passivity property of the Cartesian impedance controller is violated. Hence, if a stable interaction with the robot has to be guaranteed,

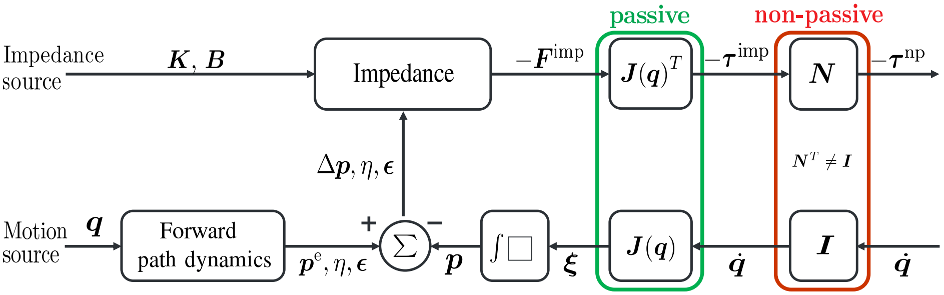


Figure 3.3: Conceptual block diagram of a Cartesian impedance controller with separated paths for forward path dynamics and impedance (inspired by Hogan (2014) and Dietrich et al. (2016)). The output of the simple impedance controller is $-\boldsymbol{\tau}^{\text{imp}}$. This torque is modified by the nullspace projector $\boldsymbol{N}$. Since the motion input is not modified by the nullspace projection method, power continuity is not preserved.

controller extensions have to be provided to ensure the stability of the robot, e.g., via energy tank methods (Dietrich et al., 2016, 2017; Garofalo and Ott, 2018; Shahriari et al., 2018, 2020).

3.3 Detection of task conflicts

To be effective, a robot must always fulfill its main task (superscript 1). For a robot with many DOF, additional tasks (task 2 ... task i) can be assigned that can be used to select the robot configuration, perhaps to optimize its kinematic and dynamic properties during the process (Ajoudani et al., 2015; Busson et al., 2017). In the rest of the chapter, a conflict between a main task and a subordinate task or a conflict between subordinate tasks with different priorities is called “task conflict.”

Fig. 3.4 shows a robotic massage assistant as an example application of multiple tasks. The main task is to move the robotic hand on the patient’s body. A massage force is generated by selecting a trajectory with virtual (rest) position in the patients body and by selecting the appropriate impedance parameters. The user can assign multiple subordinate tasks. One such task could be to keep the robot elbow at a fixed position in order to prevent large joint displacements that could harm the patient. Another task could be to hold an initial end-effector orientation that is selected by the physiotherapist. However, these tasks could be sacrificed if necessary in order to guarantee that the robot stays on the allowed path. For this application, the robot stability is safety-critical and has to always be guaranteed.

For applications like the one shown in fig. 3.4, it might be desirable to indicate to the user that not all tasks are compatible with the main task or with tasks of higher priority. The user therefore has to find an appropriate measure to quantify a task conflict. This chapter will use the knowledge of chapter 2 to detect kinematic singularities. The singularity measure will then be modified in order to predict

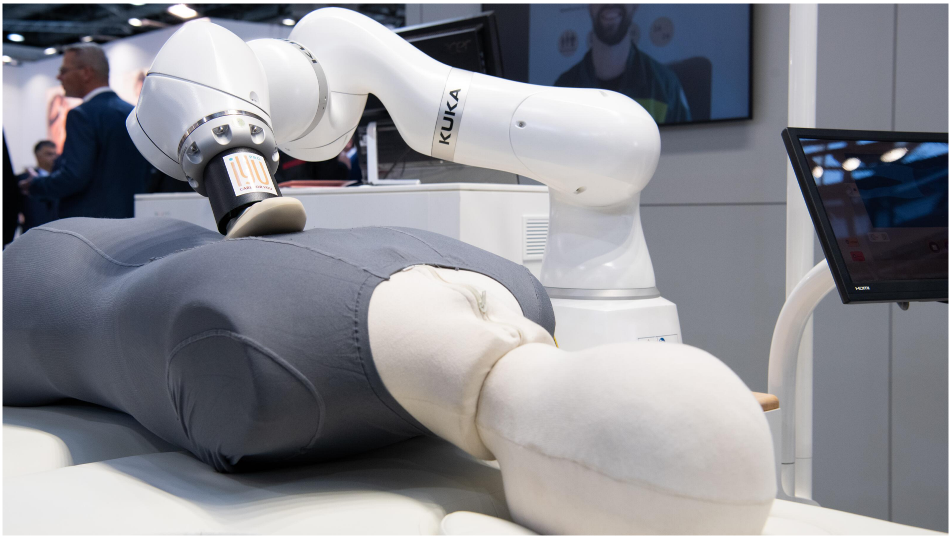


Figure 3.4: Robotic massage application developed by team iYU Pro of KUKA Innovation Award 2019 (Eyssautier and Eyssautier, 2021). The trajectory is the main task that should not be affected by additional tasks with lower priority.

task conflicts.

In sec. 2.3 the quadratic form $\boldsymbol{\Lambda}(\boldsymbol{q})^{-1}$ is derived (fig. 2.4 and eq. (2.18)). With respect to the chosen reference point, the matrix $\boldsymbol{\Lambda}(\boldsymbol{q})^{-1}$ is a block matrix and can be partitioned into four blocks: two 3×3 matrices describing the coupling terms between translational and rotational components; the matrix $\boldsymbol{\Lambda}_p(\boldsymbol{q})^{-1} \in \mathbb{R}^{3\times 3}$ mapping linear momentum to linear velocities; the matrix $\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q})^{-1} \in \mathbb{R}^{3\times 3}$ mapping angular momentum to angular velocities.

As shown in chapter 2, $\boldsymbol{\Lambda}(\boldsymbol{q})^{-1}$ is a good candidate for manipulability analysis, i.e., to detect kinematic singularities. Near singular configurations, the singular values of $\boldsymbol{\Lambda}(\boldsymbol{q})^{-1}$ go to zero.

Eq. (3.16) can be extended to:

$$\boldsymbol{\tau}^{\mathrm{np}} = \boldsymbol{\tau}^1 + \underbrace{\boldsymbol{N}^1 \; \boldsymbol{J}^2(\boldsymbol{q})^T}_{(\boldsymbol{J}^2(\boldsymbol{q})(\boldsymbol{N}^1)^T)^T} \; \boldsymbol{F}^2 \tag{3.17}$$

Here, $\boldsymbol{J}^2(\boldsymbol{q})^T$ of a subordinate task is multiplied with the nullspace projection matrix $\boldsymbol{N}^1$ of the main task (eq. (3.15)). The range of the matrix

$$\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})^T = \boldsymbol{N}^1 \; \boldsymbol{J}^2(\boldsymbol{q})^T \tag{3.18}$$

is the space of all torques that balance subordinate task wrenches without affecting the main task. The superscript “2c” should indicate the consistency of the subordinate task with the main task. Similar, the range of the matrix

$$\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q}) = \boldsymbol{J}^2(\boldsymbol{q})(\boldsymbol{N}^1)^T \tag{3.19}$$

is the space of all subordinate task motions that are consistent with the task motions of the main task (Siciliano and Slotine, 1991; Khatib and Sentis, 2004; Sentis and Khatib, 2005, 2006).

A task conflict can be detected by analyzing the rank of $\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})$. During a task conflict, the matrix $\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})$ becomes rank-deficient. The matrix $\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})$ represents a kinematic characteristic, i.e., the mapping of joint motions to task motions that are consistent with task motions of higher priority. For torque controlled robots, the ability of a robot to perform multiple tasks is dependent also on the inertial properties of the robot. In chapter 2, the mobility end-point tensor (eq. (2.18)) was analyzed to detect kinematic singularities, i.e., conflicts in a single task. This matrix can be modified such that the inertial properties of a subordinate task are decoupled from the inertial properties of the main task (Sentis and Khatib, 2005). Therefore, $\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})$ can be used to derive the quadratic form:

$$\boldsymbol{\Lambda}^{2\mathrm{c}}(\boldsymbol{q})^{-1} = \boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})\ \boldsymbol{M}(\boldsymbol{q})^{-1}\ \boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})^{T} \in \mathbb{R}^{6\times 6}. \tag{3.20}$$

The matrices $\boldsymbol{J}^{2\mathrm{c}}(\boldsymbol{q})$ and $\boldsymbol{\Lambda}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ also suffer from kinematic singularities. The rank of these matrices is now additionally influenced by the nullspace projector $\boldsymbol{N}^1$. In the following, $\boldsymbol{\Lambda}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ will be analyzed to quantify a task conflict (Sentis and Khatib, 2005, 2006; Schettino et al., 2021).

With respect to the chosen reference point, the matrix $\boldsymbol{\Lambda}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ is also a block matrix. The matrix $\boldsymbol{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1} \in \mathbb{R}^{3\times 3}$ maps linear momentum to consistent linear velocities and the matrix $\boldsymbol{\Lambda}_\epsilon^{2\mathrm{c}}(\boldsymbol{q})^{-1} \in \mathbb{R}^{3\times 3}$ maps angular momentum to consistent angular velocities.

As can be seen in eq. (3.8), the impedance potential energy is separated into $\mathscr{U}_p$ with translational stiffness $\boldsymbol{K}_p$ and $\mathscr{U}_\epsilon$ with rotational stiffness $\boldsymbol{K}_\epsilon$. Since these two impedances should be separately modified during task execution, it makes sense to also separate the task conflict detection by analyzing $\boldsymbol{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ and $\boldsymbol{\Lambda}_\epsilon^{2\mathrm{c}}(\boldsymbol{q})^{-1}$, respectively.

3.4 Shaping impedance to comply with constrained task dynamics

As can be seen in fig. 3.3, the impedance source can be specified independently of the forward path dynamics. Usually, the impedance parameters are initialized at the beginning of the application and stay constant during task execution. For multiple tasks, however, the impedances have to be adapted during task conflicts. This chapter presents an algorithm that shapes the impedances according to constrained task dynamics along/about specific directions. The conflict detection is separated for translational and rotational task dynamics, represented by $\boldsymbol{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ and $\boldsymbol{\Lambda}_\epsilon^{2\mathrm{c}}(\boldsymbol{q})^{-1}$. In case of a task conflict, the stiffnesses $\boldsymbol{K}_p$ and $\boldsymbol{K}_\epsilon$ are scaled down to comply with tasks of higher priority. The derived damping design method takes into account the modified stiffnesses. Moreover, the damping design method is extended to handle joint motions that do not affect the task motions. The impedance shaping approach modifies the elastic potential in the virtual springs. The presented method ensures that the passivity of the impedance controller is not violated (def. 3.1). The impedance optimization can be executed in parallel to the forward path dynamics, which has computational advantages. This will be shown later in the experiments on a real robot (sec. 3.5).

3.4.1 Incorporating constrained task masses

For a given unit direction $\boldsymbol{u}_j \in \mathbb{R}^3$, the *constrained task mass* (CTM) can be calculated by:

$$\mathbb{R} \ni m_j^{\text{con}} = \left(\boldsymbol{u}_j^T\ \boldsymbol{\Lambda}_p^{2\text{c}}(\boldsymbol{q})^{-1}\ \boldsymbol{u}_j\right)^{-1}. \tag{3.21}$$

The unit vectors $\boldsymbol{u}_j$ are chosen to be the three principal vectors[2] of $\boldsymbol{\Lambda}_p^{2\text{c}}(\boldsymbol{q})^{-1}$, hence $j = \{1, 2, 3\}$. The magnitude of m_j^{con} gives information about a task conflict along the direction $\boldsymbol{u}_j$: A low value of m_j^{con} will show a feasible task direction, while a high value of m_j^{con} shows that a high CTM is counteracting the task (Sentis and Khatib, 2006). This chapter will present one way to determine a high or low value. At least for industrial robots, a relation between available payload and robot weight exists. While a CTM of 8 kg might show a task conflict for a lightweight robot, this value might be low for a robot with a payload of 1000 kg. Hence, a translational task conflict can be defined by comparing m_j^{con} with the weight of the robot $m^{\text{rob}} \in \mathbb{R}$. In case of a task conflict:

$$m_j^{\text{con}} > a\ m^{\text{rob}}. \tag{3.22}$$

Here, $a > 0$ being the scalar weight-ratio for the conflict detection. The choice for an appropriate weight-ratio will be illustrated in sec. 3.5.1 and discussed in sec. 3.6. Since $\boldsymbol{\Lambda}_p^{2\text{c}}(\boldsymbol{q})^{-1}$ incorporates mass information of the robot at hand, the measure can be used for a lightweight robot like the KUKA LBR iiwa, as well as for a high-payload robot like the KUKA KR 1000 titan.

In case of a task conflict (eq. (3.22)), the task stiffnesses have to be modified in order to avoid accelerations along conflicting directions. Since the coordinates of the translational stiffness $\boldsymbol{K}_p$ and $\boldsymbol{\Lambda}_p^{2\text{c}}(\boldsymbol{q})^{-1}$ are usually not aligned, coordinates have to be transformed. For each unit vector $\boldsymbol{u}_j$ with respective CTM, a scalar stiffness $k_{u_j} \in \mathbb{R}$ is produced:

$$k_{u_j} = \boldsymbol{u}_j^T\ \boldsymbol{K}_p\ \boldsymbol{u}_j. \tag{3.23}$$

If a task conflict occurs (eq. (3.22)), a new task stiffness $k'_{u_j} = \gamma\ k_{u_j} \in \mathbb{R}$ will be assigned by using the scaling function $\gamma : \mathbb{R} \to \mathbb{R}$:

$$\gamma = \begin{cases} 1 & \text{, if } m_j^{\text{con}} \leq a\ m^{\text{rob}} \\ \dfrac{a\ m^{\text{rob}}}{m_j^{\text{con}}} & \text{, else.} \end{cases} \tag{3.24}$$

[2] For $\boldsymbol{\Lambda}_p^{2\text{c}}(\boldsymbol{q})^{-1}$ the right and left principal vectors are the same.

Since the lower condition of eq. (3.24) is always < 1 during a task conflict, k_{u_j} will be scaled down in order to comply with the main task. As the last step, the translational task stiffness has to be transformed back to align with the original coordinates of $\boldsymbol{K}_p$. This will produce a new stiffness matrix $\boldsymbol{K}'_p \in \mathbb{R}^{3\times 3}$:

$$\boldsymbol{K}'_p = \sum_{i=1}^{3} \boldsymbol{u}_j \; k'_{u_j} \; \boldsymbol{u}_j^T. \tag{3.25}$$

In the rest of the chapter, all variables rendered time-varying through the impedance shaping approach will be indicated with a prime superscript ($\prime$). An overview of the stiffness scaling approach is illustrated in fig. 3.5. With the scaling approach, the translational part of the initial task potential (eq. (3.8)) changes to:

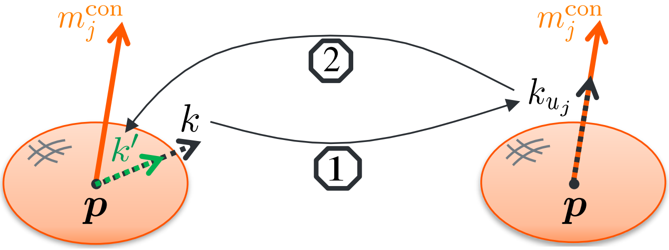


Figure 3.5: The process of scaling: The directions of m_j^{con} and k are usually not aligned (left side of figure). In the first step, the task stiffness is transformed using the principal directions of $\boldsymbol{\Lambda}_p^{2c}(\boldsymbol{q})^{-1}$. This yields the scalar stiffness k_{u_j} along u_j (right side of figure). In case of a conflict, represented by a high CTM value, the task stiffness k_{u_j} is scaled down. Afterwards, the task stiffness is transformed back to the original coordinates. This yields a new stiffness k' that is consistent with the constraint (left side of figure).

$$\mathcal{U}'_p = \frac{1}{2} \Delta \boldsymbol{p}^T \; \boldsymbol{K}'_p \; \Delta \boldsymbol{p}. \tag{3.26}$$

3.4.2 Incorporating constrained task inertias

For angular momentum $\boldsymbol{m}^1 \in \mathbb{R}^3$ that affects the main task, the angular velocity response $\boldsymbol{w}^1 \in \mathbb{R}^3$ can be represented by the equation

$$\boldsymbol{w}^1 = \boldsymbol{\Lambda}_\epsilon^1(\boldsymbol{q})^{-1} \; \boldsymbol{m}^1, \tag{3.27}$$

where $\boldsymbol{\Lambda}_\epsilon^1(\boldsymbol{q})^{-1}$ can be interpreted as an ellipsoid. The derivation and application of this ellipsoid was shown in chapter 2 (fig. 2.6).

Inverting the singular values of $\boldsymbol{\Lambda}_\epsilon^1(\boldsymbol{q})^{-1}$ yields the three inertia values I_l^1 ($l = \{1, 2, 3\}$). Fig. 3.6 shows an ellipsoid that describes the ability of an end-effector to produce motion (blue ellipsoid). The semi-axis lengths of this ellipsoid are determined by the magnitudes of I_l^1. The axes point along the principal vectors of $\boldsymbol{\Lambda}_\epsilon^1(\boldsymbol{q})^{-1}$.

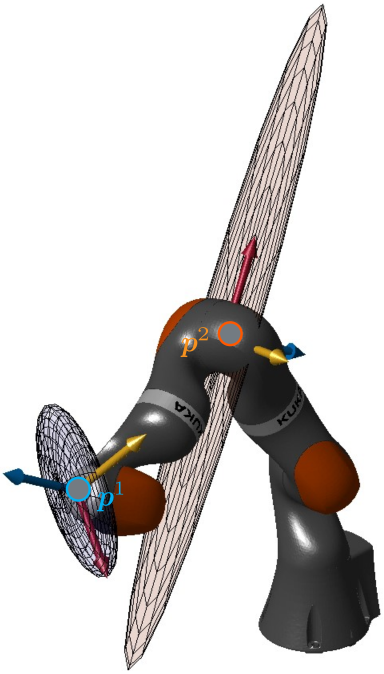


Figure 3.6: Rotational manipulability ellipsoids for a main task (a) and subordinate task that is consistent with the main task (b). (a) The blue ellipsoid shows the inertia values counteracting the main task. For representation, the inertia values were multiplied by a factor of 5. Since the robot could rotate easily about all three principal axes, low inertia values arose. (b) The orange ellipsoid represents the consistency of the subordinate task with the main task. For representation, the CTI-values were divided by a factor of 600. Since it was possible to maintain the main task positions while moving about the yellow and blue principal axes, low CTI-values arose. Since the robot could not perform rotations about the red principal axis, high CTI-values arose.

For the rotational subordinate task (superscript 2), the manipulability measure is modified to predict task conflicts: $\boldsymbol{\Lambda}_{\epsilon}^{2}(\boldsymbol{q})^{-1} \rightarrow \boldsymbol{\Lambda}_{\epsilon}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ (eq. (3.20)). With the three principal vectors[3] of $\boldsymbol{\Lambda}_{\epsilon}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$, represented by $\boldsymbol{u}_r \in \mathbb{R}^3$, the *constrained task inertia* (CTI) can be calculated:

$$\mathbb{R} \ni I_r^{\mathrm{con}} = \left(\boldsymbol{u}_r^T \ \boldsymbol{\Lambda}_{\epsilon}^{2\mathrm{c}}(\boldsymbol{q})^{-1} \ \boldsymbol{u}_r\right)^{-1}. \tag{3.28}$$

In sec. 3.4.1, it is shown that the CTM can be related to the scalar weight of the robot. Since the inertia values of a robot are usually different for each direction it is not possible to determine a direction-independent threshold.

In this chapter, a rotational task conflict is defined to occur if:

$$I_r^{\mathrm{con}} > b \ I_{\min}^{\mathrm{con}}, \tag{3.29}$$

[3] For $\boldsymbol{\Lambda}_{\epsilon}^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ the right and left principal vectors are the same.

with $b > 1$ being a threshold scalar eccentricity[4] of an ellipsoid that represents the CTI. In fig. 3.6, an ellipsoid can be seen that shows the consistency of a subordinate elbow task with a main task related to the end-effector (orange ellipsoid). Here, I_r^{con} is equal to one semi-axis length of the ellipsoid. $I_{\text{min}}^{\text{con}} \in \mathbb{R}$ is the length of the minor semi-axis. An eccentric ellipsoid shows that the subordinate task is affecting the main task. The major semi-axis shows directions with high CTI, counteracting the subordinate task. During a kinematic singularity or a task conflict, the magnitude of I_r^{con} about the conflicting direction becomes unbounded. The choice for an appropriate eccentricity threshold will be illustrated in sec. 3.5.2 and discussed in sec. 3.6.

For the given kinematic structure and the given end-effector position $\boldsymbol{p}^1 \in \mathbb{R}^3$ in fig. 3.6, the elbow position $\boldsymbol{p}^2 \in \mathbb{R}^3$ can be easily manipulated about the yellow principal axis, i.e., by rotating joints two and six. Hence, a small CTI about this direction arises. To rotate the elbow position around the blue principal axis, higher inertia values counteract the task since the robot has to move many joints. The major semi-axis (red principal axis) points along a singular direction and hence the elbow cannot be manipulated about this direction.

As in sec. 3.4.1, coordinates are transformed such that $\boldsymbol{K}_\epsilon$ is aligned with the principal directions of $\boldsymbol{\Lambda}_\epsilon^{2\text{c}}(\boldsymbol{q})^{-1}$. This yields the scalar stiffness $k_{u_r} \in \mathbb{R}$ about the direction $\boldsymbol{u}_r$:

$$k_{u_r} = \boldsymbol{u}_r^T \; \boldsymbol{K}_\epsilon \; \boldsymbol{u}_r. \tag{3.30}$$

If a task conflict occurs, a new task stiffness $\mathbb{R} \ni k'_{u_r} = \mu \; k_{u_r}$ is produced using the scaling function $\mu : \mathbb{R} \to \mathbb{R}$:

$$\mu = \begin{cases} \dfrac{b \; I_{\text{min}}^{\text{con}}}{I_r^{\text{con}}} & \text{, if } \dfrac{I_r^{\text{con}}}{I_{\text{min}}^{\text{con}}} > b \\ 1 & \text{, else.} \end{cases} \tag{3.31}$$

In the upper condition of eq. (3.31), it can be seen that for all directions where the CTI is b times bigger than the minimal CTI, the rotational stiffness k_{u_r} about this direction will be scaled down. Lastly, the rotational task stiffness is transformed back to the original coordinates to yield a new stiffness matrix $\boldsymbol{K}'_\epsilon \in \mathbb{R}^{3\times3}$:

$$\boldsymbol{K}'_\epsilon = \sum_{k=1}^{3} \boldsymbol{u}_r \; k'_{u_r} \; \boldsymbol{u}_r^T. \tag{3.32}$$

With this scaling, the rotational part of the initial task potential (eq. (3.8)) changes to:

$$\mathscr{U}'_\epsilon = 2 \; \boldsymbol{\epsilon}^T \; \boldsymbol{K}'_\epsilon \; \boldsymbol{\epsilon}. \tag{3.33}$$

If no task conflict appears (upper condition of eq. (3.24) and lower condition of eq. (3.31)), $\boldsymbol{K}_p = \boldsymbol{K}'_p$ and $\boldsymbol{K}_\epsilon = \boldsymbol{K}'_\epsilon$. During a task conflict, the matrices $\boldsymbol{K}'_p$ and $\boldsymbol{K}'_\epsilon$ are not necessarily diagonal anymore. However, these matrices are symmetric

[4] Ratio between major and minor semi-axes.

and positive-definite, which is a necessary feature for the damping design method in the next subchapter (Ott, 2008).

With $\mathscr{U}_p'$ (eq. (3.26)) and $\mathscr{U}_\epsilon'$ (eq. (3.33)), the control wrench $(\boldsymbol{F}^{\text{imp}})'$ can be calculated based on appendix D.

3.4.3 Damping design

The damping design in this subsection is split in two parts: Firstly, a damping design method is shown that can be used for a main task and a subordinate task; Secondly, this method is extended to control the remaining joint motions of the robot that do not affect the task dynamics of the main task and the subordinate task.

In robotics, constant damping matrices are often used which have a negative effect on the robot's performance. A more effective method is to express the translational and rotational damping matrices $\boldsymbol{B}_p \in \mathbb{R}^{3\times3}$ and $\boldsymbol{B}_\epsilon \in \mathbb{R}^{3\times3}$ as functions of the desired compliant behavior, represented by $\boldsymbol{K}_p'$ and $\boldsymbol{K}_\epsilon'$, and take into consideration inertial information about the robot, represented by $\boldsymbol{\Lambda}_p(\boldsymbol{q}) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q}) \in \mathbb{R}^{3\times3}$. The damping design can be derived by formulating the homogeneous equation of the error dynamics (Albu-Schaffer et al., 2003). The singular vectors of $\boldsymbol{B}_p$ and $\boldsymbol{B}_\epsilon$ are then aligned with those of $\boldsymbol{K}_p'$, $\boldsymbol{K}_\epsilon'$, $\boldsymbol{\Lambda}_p(\boldsymbol{q})$ and $\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q})$. A damping coefficient $\zeta \in \mathbb{R}$ can be defined to have a (sub-)critical damping for each singular direction. Modifying the method in Albu-Schaffer et al. (2003), the damping matrices are calculated by:

$$\boldsymbol{B}_p = \sqrt{\boldsymbol{\Lambda}_p(\boldsymbol{q})}\boldsymbol{D}_\zeta\sqrt{\boldsymbol{K}_p'} + \sqrt{\boldsymbol{K}_p'}\boldsymbol{D}_\zeta\sqrt{\boldsymbol{\Lambda}_p(\boldsymbol{q})}\ , \tag{3.34a}$$

$$\boldsymbol{B}_\epsilon = \sqrt{\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q})}\boldsymbol{D}_\zeta\sqrt{\boldsymbol{K}_\epsilon'} + \sqrt{\boldsymbol{K}_\epsilon'}\boldsymbol{D}_\zeta\sqrt{\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q})}\ , \tag{3.34b}$$

where $\boldsymbol{D}_\zeta = diag\{\zeta\}$, with $0 \leq \zeta \leq 1$ determining the desired damping behavior. The square roots of the matrices in eq. (3.34) can be calculated by taking the square roots of each singular value of the respective matrix. Afterwards, the matrices are transformed back to original coordinates. It is worth mentioning that a threshold for the singular values of $\boldsymbol{\Lambda}_p(\boldsymbol{q})$ and $\boldsymbol{\Lambda}_\epsilon(\boldsymbol{q})$ has to be set since some singular values grow without bound as singular configurations are approached. The damping design of eq. (3.34) has the advantage that the damping behavior is automatically adapted if the task stiffness is scaled during a conflict.

The passivity condition of def. 3.1 only guarantees passivity of the controller with respect to the interaction port but does not take into account internal energy exchange of the robot due to joint motion. This is especially important for kinematically redundant robots for which a nullspace in $\boldsymbol{J}(\boldsymbol{q})$ exists. Hence, joint damping has to be included to avoid oscillations.

Assigning a fixed set of joint dampers will usually conflict with the main task and subordinate tasks. Therefore, a joint damping design method has to be found that does not affect the desired workspace behavior. For a given damping matrix

$\boldsymbol{B} \in \mathbb{R}^{6\times 6}$ that incorporates the damping matrices of eq. (3.34):

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_p & 0 \\ 0 & \boldsymbol{B}_\epsilon \end{pmatrix}, \tag{3.35}$$

this matrix can be mapped into the joint space:

$$\boldsymbol{B}_q = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{B}\ \boldsymbol{J}(\boldsymbol{q}). \tag{3.36}$$

Since $\boldsymbol{B}$ is physically a (twice covariant) tensor, the mapping of eq. (3.36) always exists (Frankel, 2012). One can use the mapping of eq. (3.36) for a main task as well as for a subordinate task (indicated with superscript 1 and 2, respectively). For the resulting matrices $\boldsymbol{B}_q^1$ and $\boldsymbol{B}_q^2$, both joint damping matrices can be superimposed:

$$\boldsymbol{B}_q^{\mathrm{su}} = \boldsymbol{B}_q^1 + \boldsymbol{B}_q^2. \tag{3.37}$$

If two matrices with different rank are multiplied, the result is equal to the minimum of both ranks. By analyzing eq. (3.36), it can be seen that the matrix $\boldsymbol{B}$ is of rank six and the matrix $\boldsymbol{J}(\boldsymbol{q})$ is of rank $r \leq 6$. Therefore, the matrix $\boldsymbol{B}_q$ is also of rank r. Through the superposition of the damping matrices $\boldsymbol{B}_q^1$ and $\boldsymbol{B}_q^2$ (eq. (3.37)), the rank of matrix $\boldsymbol{B}_q^{\mathrm{su}}$ is not necessarily equal to the highest rank of its superimposed components. It turns out that through the superposition, the rank of $\boldsymbol{B}_q^{\mathrm{su}}$ is $r^{\max} \leq n$. This leaves a $(n - r^{\max})$-dimensional nullspace, which can be revealed by diagonalizing the symmetric matrix $\boldsymbol{B}_q^{\mathrm{su}}$ with the orthogonal matrix $\boldsymbol{U}_B \in \mathbb{R}^{n\times n}$:

$$\bar{\boldsymbol{B}}_q^{\mathrm{su}} = \boldsymbol{U}_B^T\ \boldsymbol{B}_q^{\mathrm{su}}\ \boldsymbol{U}_B = \begin{pmatrix} \beta_1 & ... & 0 \\ 0 & \ddots & 0 \\ 0 & ... & \beta_n \end{pmatrix}. \tag{3.38}$$

The number of zero values on the main diagonal of $\bar{\boldsymbol{B}}_{\boldsymbol{q}}^{\mathrm{su}}$ is equal to the dimension of the nullspace. To control this nullspace, a nullspace damper $\beta^{\mathrm{ns}} \in \mathbb{R}^+$ is assigned for each zero entry in $\bar{\boldsymbol{B}}_q^{\mathrm{su}}$:

$$\beta_s = \begin{cases} \beta_s & \text{, if } \beta_s > 0 \\ \beta^{\mathrm{ns}} & \text{, else.} \end{cases} \tag{3.39}$$

This is done for $1 \leq s \leq n$. By selecting appropriate values for β^{ns}, the user can influence the dynamic response to external forces, e.g., during pHRI applications. For $0 < \beta^{\mathrm{ns}} < 1$, the robot can be moved along the directions indicated by those columns of $\boldsymbol{U}_B$ that are associated with zeros on the main diagonal of $\bar{\boldsymbol{B}}_q^{\mathrm{su}}$. For $\beta^{\mathrm{ns}} > 1$, the robot will show a highly damped behavior.

As a last step, the resulting matrix is transformed back to the original coordinates:

$$\boldsymbol{B}_q{}' = \boldsymbol{U}_B\ \bar{\boldsymbol{B}}_q^{\mathrm{su}}\ \boldsymbol{U}_B^T. \tag{3.40}$$

Note that this method will automatically identify a nullspace. If no nullspace exists, $\boldsymbol{B}_q{}' = \boldsymbol{B}_q^{\mathrm{su}}$. With the damping design method, the Rayleigh function changes to:

$$\mathscr{R}(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}_q{}'\ \dot{\boldsymbol{q}}. \tag{3.41}$$

3.4.4 Extension to multi-task control

So far the impedance shaping method of secs. 3.4.1 and 3.4.2 handles one subordinate task in order to comply with the main task. A robot with many DOF, however, can handle multiple tasks (Scheurer et al., 2016). These tasks may again have different priorities. For i-number of tasks with the successive nullspace projection method, eq. (3.17) can be extended:

$$\boldsymbol{\tau}^{\mathrm{np}} = \boldsymbol{\tau}^1 + \boldsymbol{N}^1(\boldsymbol{\tau}^2 + \boldsymbol{N}^2(\boldsymbol{\tau}^3 + ... + \boldsymbol{N}^{i-1}(\boldsymbol{J}^i(\boldsymbol{q})^T \boldsymbol{F}^i \ )...)). \tag{3.42}$$

The control torque of a task i that is consistent with $i-1$ tasks with higher priority can be calculated with:

$$\boldsymbol{\tau}^{ic} = \underbrace{(\prod_{j=1}^{i-1} \boldsymbol{N}^j)\boldsymbol{J}^i(\boldsymbol{q})^T}_{(\boldsymbol{J}^i(\boldsymbol{q})(\prod_{j=1}^{i-1}\boldsymbol{N}^j)^T)^T} \ \boldsymbol{F}^i. \tag{3.43}$$

Here, the control torque of task i is multiplied with the product of all nullspace projection matrices $\boldsymbol{N}^1\boldsymbol{N}^2...\boldsymbol{N}^{i-1}$ of higher priority. The range of the matrix

$$\boldsymbol{J}^{ic}(\boldsymbol{q}) = \boldsymbol{J}^i(\boldsymbol{q}) \ (\prod_{j=1}^{i-1} \boldsymbol{N}^j)^T \tag{3.44}$$

is the space of all subordinate task motions that are consistent with $i-1$ task motions of higher priority (Sentis and Khatib, 2005, 2006). Consequently, the quadratic form $\boldsymbol{\Lambda}^{ic}(\boldsymbol{q})^{-1} \in \mathbb{R}^{6\times 6}$ can be calculated that is consistent with $i-1$ tasks:

$$\boldsymbol{\Lambda}^{ic}(\boldsymbol{q})^{-1} = \boldsymbol{J}^{ic}(\boldsymbol{q}) \ \boldsymbol{M}(\boldsymbol{q})^{-1} \ \boldsymbol{J}^{ic}(\boldsymbol{q})^T. \tag{3.45}$$

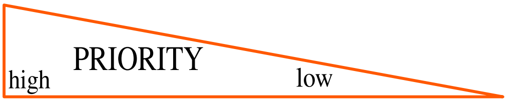
With respect to the body-fixed coordinate frame, $\boldsymbol{\Lambda}^{ic}(\boldsymbol{q})^{-1}$ is again a block matrix which incorporates the 3×3-matrices $\boldsymbol{\Lambda}_p^{ic}(\boldsymbol{q})^{-1}$ and $\boldsymbol{\Lambda}_\epsilon^{ic}(\boldsymbol{q})^{-1}$. With these two matrices a conflict can again be separately detected for translational and rotational tasks. If a conflict is detected, the scaling method of secs. 3.4.1 and/or 3.4.2 and the damping method of sec. 3.4.3 can be performed. Finally, the control torques resulting from all individual impedances can be superimposed, which yields $\boldsymbol{\tau}' \in \mathbb{R}^n$.

3.4.5 Comparison of task hierarchy

Predictability is a very important feature of industrial robot applications. For robots that perform multiple tasks, it is therefore desired that the robot indicates that not all tasks are mutually compatible (sec. 3.1.2). The task conflict detection of eqs. (3.22) and (3.29) can be used for all multi-task control approaches, e.g., for impedance superposition and nullspace projection.

For the successive nullspace projection approach, the priority assignment can be achieved by nested summation of tasks, expressed as torques, with different priorities (eq. (3.42)). The damping torque that controls the motions in the nullspace of $\boldsymbol{J}^i(\boldsymbol{q})$ (eq. (3.43)) is the lowest priority and will be multiplied with the product of all nullspace projection matrices $\boldsymbol{N}^1\boldsymbol{N}^2...\boldsymbol{N}^i$ that are associated to the preceding tasks.

For the impedance shaping method, the task conflict detection and the control reaction are unified in one approach. The task hierarchy is implicitly assigned through the impedance shaping method (secs. 3.4.1 and 3.4.2). The resulting task torques with different task priority are superimposed. The damping design takes into account the modified stiffnesses during a task conflict. Moreover, the damping design automatically handles joint motions that do not affect the task dynamics. Compared to the nullspace projection method, this is done without explicitly assigning an additional (lowest) priority. The priority assignment can be visualized as a triangle (fig. 3.7). While the main task has the highest priority, subordinate tasks may or may not be feasible without affecting the main task or tasks with higher priority.



$$\boldsymbol{\tau}' \; = \boldsymbol{J}^1(\boldsymbol{q})^T \boldsymbol{F}^{\mathrm{imp},1} \; + \; ... \qquad ... \; + \; \boldsymbol{J}^i(\boldsymbol{q})^T (\boldsymbol{F}^{\mathrm{imp},i})' \; - \; \boldsymbol{B}'_q \dot{\boldsymbol{q}}$$

Figure 3.7: Triangle of priority. From left to right, the priority of different tasks is listed in descending order. The control wrenches are calculated based on appendix D. While the main task always has to be respected, subordinate tasks may be sacrificed and therefore have lower priority. The damping task incorporates the desired Cartesian damping behavior for all task hierarchies.

3.4.6 Hierarchical passivity check

To make a statement about the overall passivity of the controller, each physically interpretable and power continuously connected subsystem of the Cartesian impedance controller can be analyzed separately. Since torque controlled robots with revolute joints are considered in this chapter, the gravity compensation is lossless with respect to $\left(\dot{\boldsymbol{q}}, \boldsymbol{g}\right)$. Hence, the subsystem to compensate for gravitational effects does not have to be handled in the further analysis.

Through the scaling approach, the controller has additional time-varying components. If the controller handles one main task and one subordinate task, these additional time-varying components are $\boldsymbol{K}_p^{2'}$ and $\boldsymbol{K}_\epsilon^{2'}$. Through the scaling approach (sec. 3.4), these two matrices are not necessarily diagonal any more. Since $\boldsymbol{K}_p^{2'}$ and $\boldsymbol{K}_\epsilon^{2'}$ are symmetric, the matrices have dependent (off-diagonal) elements. To apply partial differentiation of $\mathscr{U}^{2'}(\Delta\boldsymbol{p}^2, \boldsymbol{\epsilon}^2)$ with respect to $\boldsymbol{K}_p^{2'}$ and $\boldsymbol{K}_\epsilon^{2'}$, a local parametrization of $\mathbb{R}_+^3$ has to be found, such that $\boldsymbol{K}_p^{2'}$ and $\boldsymbol{K}_\epsilon^{2'}$ are diagonal matrices. The spectral decomposition of a given stiffness matrix $\boldsymbol{K}' \in \mathbb{R}^{3\times 3}$ is:

$$\boldsymbol{K}' = \boldsymbol{R}\ \bar{\boldsymbol{K}}\ \boldsymbol{R}^T, \tag{3.46}$$

with $\boldsymbol{R} \in SO(3)$ being the rotation matrix, $\bar{\boldsymbol{K}} = \mathrm{diag}(\bar{\boldsymbol{k}}) \in \mathbb{R}^{3\times 3}$ being a diagonal positive-definite stiffness matrix, and $\bar{\boldsymbol{k}} \in \mathbb{R}^3$ being a vector that incorporates the three scalar stiffness values. If $\boldsymbol{K}'$ is time-varying, $\bar{\boldsymbol{K}}$ and $\boldsymbol{R}$ are time-varying. However, $\boldsymbol{R}$ has dependent matrix elements. To apply partial differentiation, another global representation of rotations has to be chosen. The exponential coordinates for rotation were shown in sec. 3.2. Eq. (3.6) can be rearranged to:

$$\boldsymbol{R}(\boldsymbol{\epsilon}, \eta) = \exp(\tilde{\boldsymbol{\epsilon}}\eta). \tag{3.47}$$

Here, $\tilde{\boldsymbol{\epsilon}}$ is the skew-symmetric matrix representation of the unit-axis $\boldsymbol{\epsilon} \in \mathbb{R}^3$, and $\eta \in \mathbb{R}$ is the rotation angle. To calculate $\exp(\tilde{\boldsymbol{\epsilon}}\eta)$, "Rodrigues' formula" (Rodrigues, 1816; Murray et al., 1994) can be used:

$$\exp(\tilde{\boldsymbol{\epsilon}}\eta) = \boldsymbol{I} - \tilde{\boldsymbol{\epsilon}}\ \sin(\eta) + \tilde{\boldsymbol{\epsilon}}^2(1 - \cos(\eta)). \tag{3.48}$$

With eqs. (3.46) and (3.47), the potential function $\mathscr{U}^{2'}$ assigned to a subordinate task can be represented as the sum of the translational potential $\mathscr{U}_\mathrm{p}^2(\Delta\boldsymbol{p}^2, \bar{\boldsymbol{\epsilon}}_\mathrm{p}, \bar{\eta}_\mathrm{p}, \bar{\boldsymbol{k}}_\mathrm{p}^2)$ and the rotational potential $\mathscr{U}_\epsilon^2(\boldsymbol{\epsilon}^2, \bar{\boldsymbol{\epsilon}}_\epsilon, \bar{\eta}_\epsilon, \bar{\boldsymbol{k}}_\epsilon^2)$:

$$\mathscr{U}_\mathrm{p}^2(\Delta\boldsymbol{p}^2, \bar{\boldsymbol{\epsilon}}_\mathrm{p}, \bar{\eta}_\mathrm{p}, \bar{\boldsymbol{k}}_\mathrm{p}^2) = \frac{1}{2}(\Delta\boldsymbol{p}^2)^T \exp(\tilde{\bar{\boldsymbol{\epsilon}}}_\mathrm{p}\bar{\eta}_\mathrm{p})\ \mathrm{diag}(\bar{\boldsymbol{k}}_\mathrm{p}^2)\ \exp(\tilde{\bar{\boldsymbol{\epsilon}}}_\mathrm{p}^T\bar{\eta}_\mathrm{p})\ \Delta\boldsymbol{p}^2; \tag{3.49a}$$

$$\mathscr{U}_\epsilon^2(\boldsymbol{\epsilon}^2, \bar{\boldsymbol{\epsilon}}_\epsilon, \bar{\eta}_\epsilon, \bar{\boldsymbol{k}}_\epsilon^2) = 2\ (\boldsymbol{\epsilon}^2)^T \exp(\tilde{\bar{\boldsymbol{\epsilon}}}_\epsilon\bar{\eta}_\epsilon)\ \mathrm{diag}(\bar{\boldsymbol{k}}_\epsilon^2)\ \exp(\tilde{\bar{\boldsymbol{\epsilon}}}_\epsilon^T\bar{\eta}_\epsilon)\ \boldsymbol{\epsilon}^2. \tag{3.49b}$$

Time differentiation of $S^{\mathrm{ctrl}'} = \mathscr{U}^1 + \mathscr{U}^{2'}$ yields:

$$\begin{aligned} \dot{S}^{\mathrm{ctrl}'} = {} & (\frac{\partial\mathscr{U}_p^1}{\partial\Delta\boldsymbol{p}^1})^T\ \dot{\Delta\boldsymbol{p}}^1 + (\frac{\partial\mathscr{U}_\epsilon^1}{\partial\Delta\boldsymbol{\epsilon}^1})^T\ \dot{\boldsymbol{\epsilon}}^1 \\ & + (\frac{\partial\mathscr{U}_p^{2'}}{\partial\Delta\boldsymbol{p}^2})^T\ \dot{\Delta\boldsymbol{p}}^2 + (\frac{\partial\mathscr{U}_\epsilon^{2'}}{\partial\boldsymbol{\epsilon}^2})^T\ \dot{\boldsymbol{\epsilon}}^2 \\ & \underbrace{+\ (\frac{\partial\mathscr{U}_p^{2'}}{\partial\bar{\boldsymbol{k}}_\mathrm{p}^2})^T\ \dot{\bar{\boldsymbol{k}}}_\mathrm{p}^2 + (\frac{\partial\mathscr{U}_p^{2'}}{\partial\bar{\boldsymbol{\epsilon}}_\mathrm{p}})^T\ \dot{\bar{\boldsymbol{\epsilon}}}_\mathrm{p} + (\frac{\partial\mathscr{U}_p^{2'}}{\partial\bar{\eta}_\mathrm{p}})^T\ \dot{\bar{\eta}}_\mathrm{p}}_{\dot{S}_\mathrm{p}^\star} \\ & \underbrace{+\ (\frac{\partial\mathscr{U}_\epsilon^{2'}}{\partial\bar{\boldsymbol{k}}_\epsilon^2})^T\ \dot{\bar{\boldsymbol{k}}}_\epsilon^2 + (\frac{\partial\mathscr{U}_\epsilon^{2'}}{\partial\bar{\boldsymbol{\epsilon}}_\epsilon})^T\ \dot{\bar{\boldsymbol{\epsilon}}}_\epsilon + (\frac{\partial\mathscr{U}_\epsilon^{2'}}{\partial\bar{\eta}_\epsilon})^T\ \dot{\bar{\eta}}_\epsilon}_{\dot{S}_\epsilon^\star}\,. \end{aligned} \tag{3.50}$$

The solution for the time differentiation of $\mathscr{U}_\mathrm{p}(\Delta\boldsymbol{p}, \bar{\boldsymbol{\epsilon}}_\mathrm{p}, \bar{\eta}_\mathrm{p}, \bar{\boldsymbol{k}}_\mathrm{p})$ and $\mathscr{U}_\epsilon(\boldsymbol{\epsilon}, \bar{\boldsymbol{\epsilon}}_\epsilon, \bar{\eta}_\epsilon, \bar{\boldsymbol{k}}_\epsilon)$ can be found in appendix D.3.

The sign of $\dot{S}_\mathrm{p}^\star$ and $\dot{S}_\epsilon^\star$ in the lower terms of the right side of eq. (3.50) is in general unknown. If the stiffness of the subordinate (translational or rotational) task is scaled down during a task conflict, the respective subsystem extracts energy. As long as energy is extracted, the subsystem is passive with respect to $\big(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}'\big)$. After a conflict, the stiffness is scaled up and the subsystem actively gains energy. If the subsystems actively gains energy, the controller does not necessarily have to be active with respect to port $\big(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}'\big)$. The power supplied by the controller is:

$$\begin{aligned} -\boldsymbol{\tau}'^T\ \dot{\boldsymbol{q}} = {} & (\frac{\partial \mathscr{U}_{\mathrm{p}}^1}{\partial \Delta \boldsymbol{p}^1})^T\ \dot{\Delta \boldsymbol{p}}^1 + (\frac{\partial \mathscr{U}_{\epsilon}^1}{\partial \Delta \boldsymbol{\epsilon}^1})^T\ \dot{\boldsymbol{\epsilon}}^1 \\ & + (\frac{\partial \mathscr{U}_{\mathrm{p}}^{2\prime}}{\partial \Delta \boldsymbol{p}^2})^T\ \dot{\Delta \boldsymbol{p}}^2 + (\frac{\partial \mathscr{U}_{\epsilon}^{2\prime}}{\partial \boldsymbol{\epsilon}^2})^T\ \dot{\boldsymbol{\epsilon}}^2 \\ & + \quad \underbrace{(\frac{\partial \mathscr{R}(\dot{\boldsymbol{q}})'}{\partial \dot{\boldsymbol{q}}})^T\ \dot{\boldsymbol{q}}}_{\text{positive semi-definite}} \quad . \end{aligned} \tag{3.51}$$

By comparing the passivity definition of def. 3.1 with eq. (3.50) and eq. (3.51), it can be concluded that the controller is passive (or at least lossless) with respect to the port $\big(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}'\big)$ if:

$$\dot{S}_{\mathrm{p}}^{\star} + \dot{S}_{\epsilon}^{\star} \leq (\frac{\partial \mathscr{R}(\dot{\boldsymbol{q}})'}{\partial \dot{\boldsymbol{q}}})^T\ \dot{\boldsymbol{q}}. \tag{3.52}$$

To avoid excessive energy injection through port $\big(\dot{\boldsymbol{q}}, -\boldsymbol{\tau}'\big)$, the controller has to detect and correct non-passive actions. The dissipated energy (right hand side of eq. (3.52)) is monitored and the scaling of subsystems that actively gain energy is stopped as long as the passivity condition of def. 3.1 is violated. For each control cycle, the stiffness of the previous control cycle $\boldsymbol{K}^{\mathrm{pre}\prime}$ is stored before the scaling approach of sec. 3.4 is applied. By setting $\boldsymbol{K}' = \boldsymbol{K}^{\mathrm{pre}\prime}$, the time derivative of the stiffness matrices is equal to the zero matrix and hence no more energy is injected by this subsystem.

There may be situations, however, where the dissipated energy is large enough that not all active actions have to be stopped. In this case it is desirable to handle the scaling in a hierarchical manner. The control algorithm is depicted in fig. 3.8.

3.5 Experiments on a real robot

In the experiments, the example application of fig. 3.4 will be used with two different end-effector trajectories which are considered to be the main task. An impedance controller is a promising solution for such an application since it involves pHRI. If additional subordinate tasks are assigned, multiple conflicting impedances will yield a high Cartesian error for the main task. As anticipated, nullspace projection methods should reduce this error. In this sub-section, these two traditional control methods are compared with the new impedance shaping method of sec. 3.4.

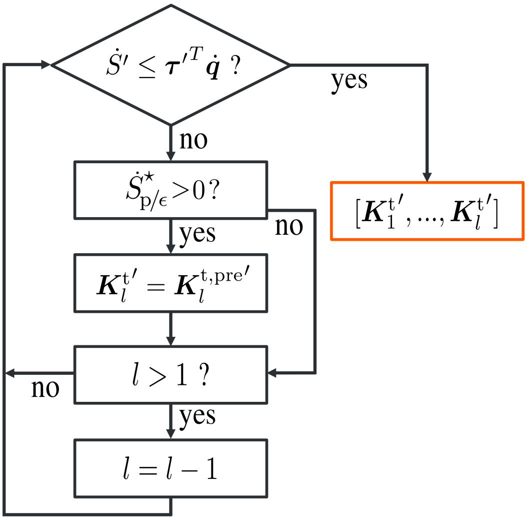

$\dot{S}' \leq \boldsymbol{\tau}'^T \dot{\boldsymbol{q}}$?
yes
no
$\dot{S}^{\star}_{\mathrm{p}/\epsilon} > 0$?
yes
no
$\boldsymbol{K}_l^{\mathrm{t}\prime} = \boldsymbol{K}_l^{\mathrm{t,pre}\prime}$
$l > 1$?
no
yes
$l = l - 1$
$[\boldsymbol{K}_1^{\mathrm{t}\prime}, ..., \boldsymbol{K}_l^{\mathrm{t}\prime}]$

Figure 3.8: Algorithm for a hierarchical passivity check to control a main task and i-number of subordinate tasks. If the passivity condition is violated, the stiffness of the active subsystem with lowest priority is not increased. The same passivity check is performed for all other tasks, proceeding from the next higher priority task to the highest priority task. The output of the algorithm is an array with stiffness matrices that guarantee passivity. These matrices are then used to calculate the impedance control torque.

For comparison, the Cartesian error $e \in \mathbb{R}$ based on the Euclidean 2-norm is calculated. Moreover, for the complete application, the root-mean-squared-error is used:

$$\mathrm{RMSE} = \sqrt{\frac{1}{ans}(e_1^2 + e_2^2 + ... + e_l^2)}, \tag{3.53}$$

with ans being the number of control cycles for each experiment.

Simple impedance superposition was based on the elastic potential approach presented in eqs. (3.8) - (3.10). For the experiments with the nullspace projection method, the successive projection method was used. For further information about implementation details of nullspace projection methods, please see Han and Park (2013).

For a simple impedance controller, the objective is to control the compliant and viscous behavior. There are also controllers that shape the desired inertial behavior, represented by $\boldsymbol{\Lambda}(\boldsymbol{q})$ (Ott, 2008). In the example of the Operational Space Control framework, the control wrench $\boldsymbol{F}^{\mathrm{imp}}$ is modified such that $\hat{\boldsymbol{F}}^{\mathrm{imp}} = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{F}^{\mathrm{imp}}$. This method, called “non-linear dynamic decoupling” in Khatib (1987), helps improve the accuracy of the robot with respect to the main task and subordinate tasks. For the nullspace projection method, this method was used. In the following, all variables for which the decoupling is applied are indicated with a hat-symbol (tab. 3.1). The advantages and disadvantages of non-linear decoupling are presented in Dietrich et al. (2021).

Two experiments with the KUKA LBR iiwa R820 were performed using the Fast-Research-torque-interface (C++) with a fixed sample rate of 5 ms (Schreiber et al., 2010). The initial configurations and the control parameters can be seen in appendices F.1 and F.2. All control torques in this section are elements of $\mathbb{R}^n$. For simplicity, the torques to compensate centrifugal/Coriolis effects and gravitational effects are left out of the equations. Fig. 3.9 shows the structure of the controller used for the experiments. The basis for all experiments was the Cartesian impedance controller. Three controller extensions were implemented. The first extension was only used for the experiments with the nullspace projection method. The second and third extensions were only used for the novel impedance shaping method. The controller was implemented on a Quad-Core Linux-PC with Intel i3-2100 processor and CPU at 3.10GHz. For the impedance shaping method, the kinematics, dynamics, and optimized impedances were computed in parallel using the C++ boost library (Demming and Duffy, 2012). For the nullspace projection method and the impedance shaping method, the computation time t_{CMP} from input data $\boldsymbol{q}, \dot{\boldsymbol{q}}$ to output command $\boldsymbol{\tau}$ was tracked using the "high resolution clock"-method of boost/chrono.

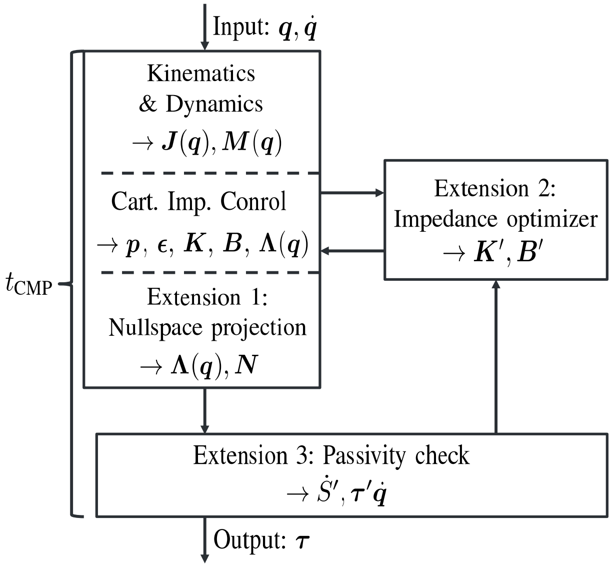

Input: $\boldsymbol{q}, \dot{\boldsymbol{q}}$
Kinematics & Dynamics
$\rightarrow \boldsymbol{J}(\boldsymbol{q}), \boldsymbol{M}(\boldsymbol{q})$
Cart. Imp. Conrol
$\rightarrow \boldsymbol{p}, \epsilon, \boldsymbol{K}, \boldsymbol{B}, \boldsymbol{\Lambda}(\boldsymbol{q})$
Extension 1: Nullspace projection
$\rightarrow \boldsymbol{\Lambda}(\boldsymbol{q}), \boldsymbol{N}$
Extension 2: Impedance optimizer
$\rightarrow \boldsymbol{K}', \boldsymbol{B}'$
t_{CMP}
Extension 3: Passivity check
$\rightarrow \dot{S}', \boldsymbol{\tau}'\dot{\boldsymbol{q}}$
Output: $\boldsymbol{\tau}$

Figure 3.9: Control structure for the impedance controller based on the elastic potential. The controller had three extensions: For the nullspace projection method the impedance torque was multiplied with the nullspace projector (extension 1); As long as the shaped impedances did not violate the passivity of the controller (extension 3), the impedances of task 2 to task i were optimized to be consistent with the main task and tasks with higher priority (extension 2).

3.5.1 Experiment 1: Move end-effector on a line and hold elbow position

For the first experiment (fig. 3.10), the robot end-effector had to move along the x-direction of the inertial coordinate frame (main task) while holding the elbow at the initial position (subordinate task). To take into account transient behavior when the robot starts moving, the end-effector motion was repeated three times back and forth. For both tasks, only a translational potential $\mathscr{U}_{\mathrm{p}}^1$ and $\mathscr{U}_{\mathrm{p}}^2$ was applied. In this subsection, the superscript i will appear in equations that are calculated for the main task and the subordinate task.

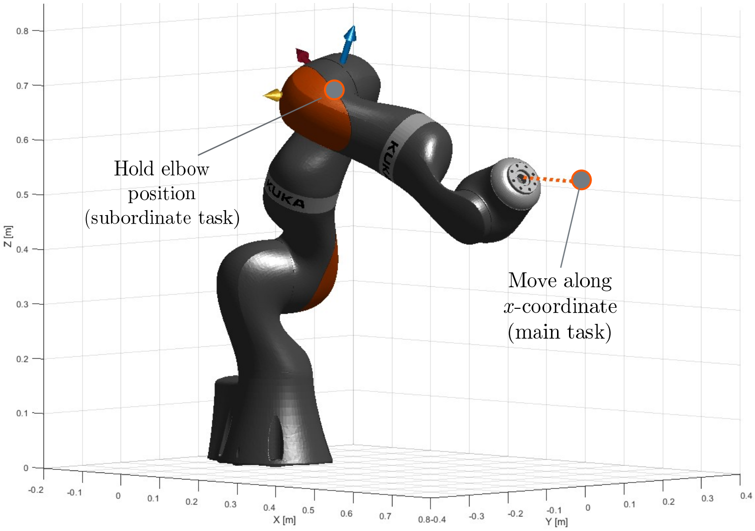


Figure 3.10: Experiment 1: The robot had to move along the x-axis of the base coordinate frame (main task) while keeping the elbow at the initial position (subordinate task). For a chosen point on the elbow body, the CTM for the task was represented along the principal vectors of $\boldsymbol{\Lambda}_p^{2c}(\boldsymbol{q})^{-1}$.

Following appendix D.1, the control torques $\boldsymbol{\tau}^i$ of the virtual translational springs were derived by computing the differential of $\mathscr{U}_p^i$ (appendix D.1):

$$\boldsymbol{\tau}^i = \boldsymbol{J}^i(\boldsymbol{q})^T \begin{pmatrix} -\boldsymbol{K}_p^i \, \Delta\boldsymbol{p}^i \\ \Delta\tilde{\boldsymbol{p}}^i \, \boldsymbol{K}_p^i \, \Delta\boldsymbol{p}^i \end{pmatrix}. \tag{3.54}$$

Here, $\Delta\tilde{\boldsymbol{p}}^i$ is the skew-symmetric matrix form of $\Delta\boldsymbol{p}^i$. For both tasks, the translational Cartesian damping matrix $\boldsymbol{B}_p^i$ was calculated based on eq. (3.34). With this matrix, the six-dimensional matrix form of eq. (3.35) was produced, where all entries of $\boldsymbol{B}_\epsilon^i$ were set to zero. Therefore, the damping torque $\boldsymbol{\tau}_B^i$ could be calculated by:

$$\boldsymbol{\tau}_B^i = \boldsymbol{J}^i(\boldsymbol{q})^T \, \boldsymbol{B}^i \, \boldsymbol{\xi}^i. \tag{3.55}$$

The joint damper torque $\boldsymbol{\tau}_{B_{\boldsymbol{q}}}$ was calculated with

$$\boldsymbol{\tau}_{B_{\boldsymbol{q}}} = \beta^{\mathrm{ns}}\ \boldsymbol{I}_n, \tag{3.56}$$

where $\beta^{\mathrm{ns}} \in \mathbb{R}$ was a scalar damper value assigned to all joints (Appendix F.1) and $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ was the identity matrix.

For simple impedance superposition, the torques of eqs. (3.54) - (3.56) were superimposed:

$$\boldsymbol{\tau}^{\mathrm{imp}} = \boldsymbol{\tau}^1 - \boldsymbol{\tau}_B^1 + \boldsymbol{\tau}^2 - \boldsymbol{\tau}_B^2 - \boldsymbol{\tau}_{B_{\boldsymbol{q}}}. \tag{3.57}$$

For the nullspace projection method, a task priority was assigned by introducing the nullspace projectors $\boldsymbol{N}^1$ and $\boldsymbol{N}^2$ which were calculated based on eq. (3.15). The final control torque for the nullspace projection method was calculated by modifying eq. (3.57):

$$\boldsymbol{\tau}^{\mathrm{np}} = \hat{\boldsymbol{\tau}}^1 - \hat{\boldsymbol{\tau}}_B^1 + \boldsymbol{N}^1\ (\hat{\boldsymbol{\tau}}^2 - \hat{\boldsymbol{\tau}}_B^2 - \boldsymbol{N}^2\ \boldsymbol{\tau}_{B_{\boldsymbol{q}}}). \tag{3.58}$$

In comparison to the nullspace projection method, the impedance shaping method implemented the task prioritization by using the optimized stiffness $\boldsymbol{K}_p^{2'}$ to calculate the task torque $\boldsymbol{\tau}^{2'}$ (sec. 3.4.1). Moreover, the novel damping design method of sec. 3.4.3 was applied which yielded the damping torque:

$$\boldsymbol{\tau}_{B_{\boldsymbol{q}}}{}' = \boldsymbol{B}_{\boldsymbol{q}}{}'\ \dot{\boldsymbol{q}}. \tag{3.59}$$

The final control torque for the impedance shaping method was:

$$\boldsymbol{\tau}' = \boldsymbol{\tau}^1 + \boldsymbol{\tau}^{2'} - \boldsymbol{\tau}_{B_{\boldsymbol{q}}}{}'. \tag{3.60}$$

While the robot was moving along the line, a task conflict appeared and the subordinate task, associated with a point on the elbow, had to be partly sacrificed. In fig. 3.11, the CTM along the three principal directions of $\boldsymbol{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1}$ can be seen. A small CTM acted along the yellow principal axis (fig. 3.10) since the elbow could be moved along this direction without affecting the main task. The CTM along the red principal axis stayed low over the experiment since the tasks are both feasible, e.g., by rotating axis 6 of the robot. Since the height of the elbow position could not be maintained while the end-effector was moving along the line, the direction along the blue principal axis was in conflict. Moreover, the blue principal axis pointed along a singular direction (cf. also fig. 3.6).

For the experiments in this chapter, a weight-ratio of a = 0.3 was set (appendix F). As can be seen in fig. 3.11, only one principal direction was in conflict and the scaling function along this directions acted on the task stiffness. For the other directions $\gamma = 1$ (eq. (3.24)). With insight derived from various experiments, the author proposes a weight-ratio of $a > 0.1$ to users of the method. A weight-ratio of $0 < a < 0.2$ will induce scaling, even for small variations of the CTM (fig. 3.11, red principal axis).

Fig. 3.12a shows the Cartesian error of both tasks. In fig. 3.12b, the RMSE can be seen. The theoretical conclusions were confirmed by this experiment: While

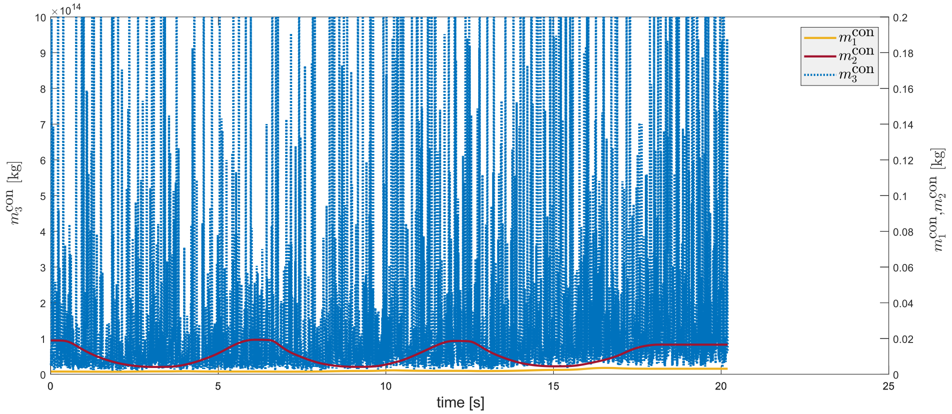
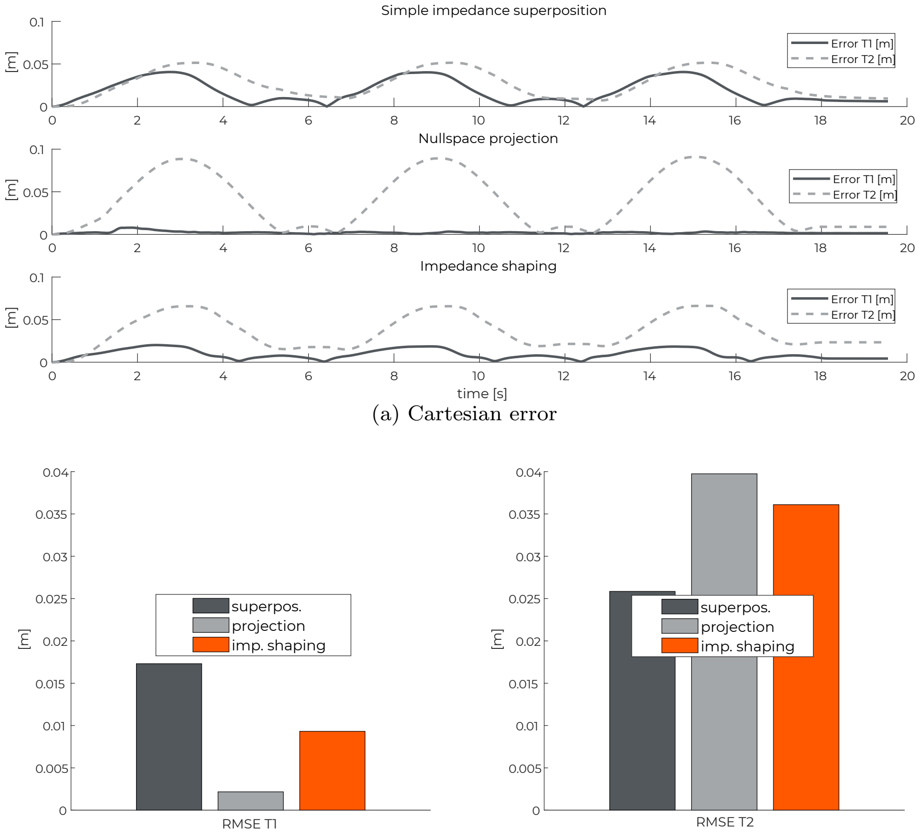


Figure 3.11: Constrained task mass (CTM) vs. time. Plot colors as in the coordinate frame in fig. 3.10, placed on the robot elbow. Along the blue axis, high CTM acted. While the CTM along the red axis varied, the CTM along the yellow axis stayed low.

simple impedance superposition showed a large Cartesian error for the main task, the nullspace projection method reduced this error significantly. Compared to simple impedance superposition, the nullspace projection method reduced the Cartesian error by 87.28%. The effectiveness of the impedance shaping method can also be seen. Compared to simple impedance superposition, the impedance shaping method reduced the Cartesian error by 46.24%. As expected, the simple impedance superposition performed best for the subordinate task since it had the same priority as the main task. The nullspace projection method and the impedance shaping method showed similar RMSE for the subordinate task (fig. 3.12b, right bar plot).



(a) Cartesian error

(b) RMSE

Figure 3.12: Cartesian error and RMSE for the main task and the subordinate elbow task.

For the impedance shaping method, extension 3 of the controller maintained passivity (def. 3.1). Fig. 3.13 shows the energy flow through the potential energy $\dot{S}^{\mathrm{ctrl}'}$ and the power supplied from the controller to the robot, calculated by $-\boldsymbol{\tau}'^{T}\dot{\boldsymbol{q}}$. As can be seen, the controller stayed passive during the experiment.

Tab. 3.2 shows the minimal, maximal, and mean computation time for the controller with the nullspace projection method (extension 1) and the novel impedance shaping method (extension 2 and 3). Due to the parallel calculation of kinematics, dynamics, and impedance optimization, the impedance shaping method reduced the mean computation time by 77.23%, compared to the nullspace projection method.

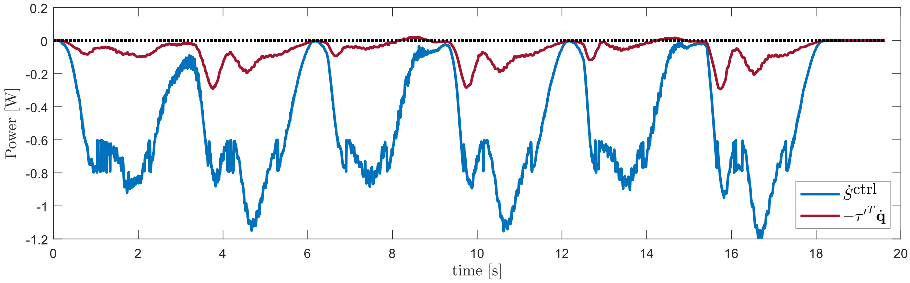


Figure 3.13: Since the supplied power was always smaller than the energy flow produced from the time-varying stiffness, the controller remained passive during the experiment. It is hypothesized that the brief epochs of positive power (near 8 seconds and 14 seconds) are due to poor friction compensation during slow joint motions.

Table 3.2: Computation time for the controller with 3926 control cycles.

$\boldsymbol{t}^{\mathrm{cmp}}$	**Nullspace projection**	**Impedance shaping**
min	0.80 ms	0.23 ms
max	1.70 ms	1.40 ms
mean	1.10 ms	0.25 ms

3.5.2 Experiment 2: Move end-effector on a circle, hold elbow position, and hold end-effector orientation

For the second experiment (fig. 3.14), the robot end-effector had to move around a circular (main task). Two additional tasks were assigned: The initial elbow-position had to be kept constant (task 2) and the initial end-effector orientation had to be maintained (task 3). Following sec. 3.4.4 and fig. 3.7, task 2 had higher priority than task 3. To take into account transient behavior, the end-effector moved four times around the circle. The three different controllers were implemented again. For the nullspace projection method, holding elbow position with high impedance magnitude in combination with the additional rotational task caused oscillations. Hence, the impedance parameters were reduced for the nullspace projection method (appendix F.2). This shows that expert knowledge is needed for the nullspace projection method, to find feasible control parameters for multiple task hierarchies.

The Jacobian matrix $\boldsymbol{J}^{3\mathrm{c}}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$, associated with task 3, was consistent with the main task and task 2 (eq. (3.44)). Since the matrix had full rank during the entire experiment, no joint damping torque was commanded.

As in experiment 1, $\boldsymbol{\tau}^1$ and $\boldsymbol{\tau}^2$ both described virtual translational springs and were calculated based on eq. (3.54). For task 3, the control torque $\boldsymbol{\tau}^3$ of the virtual

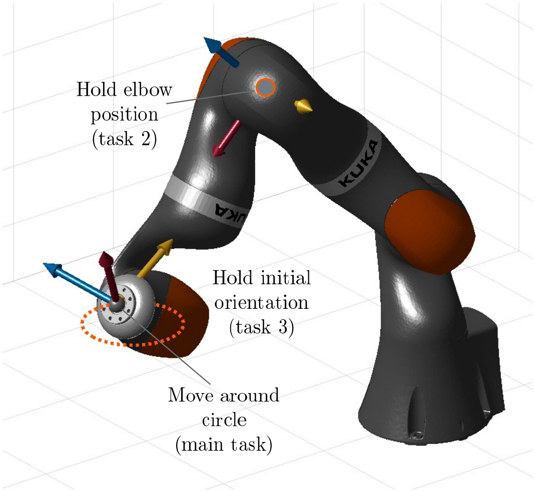


Figure 3.14: Experiment 2: The robot end-effector had to move around a circle (main task), while keeping the elbow at its initial position (task 2) and maintaining the initial end-effector orientation (task 3). For a chosen point on the elbow, the CTM was represented along the principal vectors of $\boldsymbol{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1}$. For a chosen point on the end-effector, the CTI was represented about the principal vectors of $\boldsymbol{\Lambda}_\epsilon^{3\mathrm{c}}(\boldsymbol{q})^{-1}$.

rotational spring was derived by computing the differential of $\mathscr{U}_\epsilon^3$ (appendix D.2):

$$\boldsymbol{\tau}^3 = \boldsymbol{J}^3(\boldsymbol{q})^T \begin{pmatrix} 0 \\ 2\boldsymbol{E}(\eta^3, \boldsymbol{\epsilon}^3)^T \boldsymbol{K}_\epsilon^3 \boldsymbol{\epsilon}^3 \end{pmatrix} . \tag{3.61}$$

Note that $\boldsymbol{E}(\eta^3, \boldsymbol{\epsilon}_2^{\mathrm{t}})$ can be calculated with eq. (D.11).

For the damping torques, eq. (3.55) was used again. Here, the damping design of simple impedance superposition and the nullspace projection method had the matrices $\boldsymbol{K}_p^1$, $\boldsymbol{K}_p^2$ and $\boldsymbol{K}_\epsilon^3$ as input.

For simple impedance superposition, all control torques were superimposed:

$$\boldsymbol{\tau}^{\mathrm{imp}} = \boldsymbol{\tau}^1 - \boldsymbol{\tau}_B^1 + \boldsymbol{\tau}^2 - \boldsymbol{\tau}_B^2 + \boldsymbol{\tau}^3 - \boldsymbol{\tau}_B^3. \tag{3.62}$$

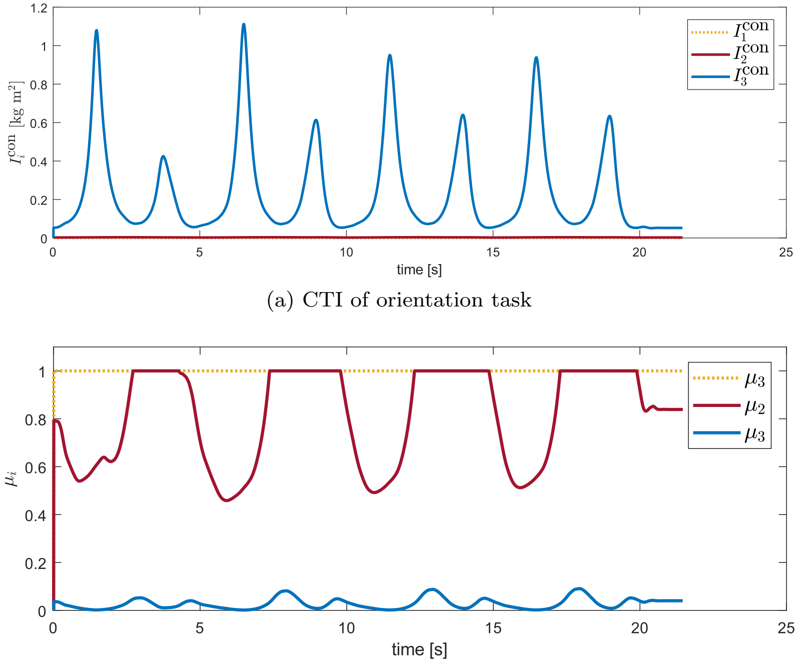
The nullspace projection method assigned a task priority by using the nullspace projectors $\boldsymbol{N}^1$ and $\boldsymbol{N}^2$ (eq. (3.15)). The final control torque $\boldsymbol{\tau}^{\mathrm{np}}$ for the nullspace projection method was:

$$\boldsymbol{\tau}^{\mathrm{np}} = \hat{\boldsymbol{\tau}}^1 - \hat{\boldsymbol{\tau}}_B^1 + \boldsymbol{N}^1 \left(\hat{\boldsymbol{\tau}}^2 - \hat{\boldsymbol{\tau}}_B^2 + \boldsymbol{N}^2(\hat{\boldsymbol{\tau}}^3 - \hat{\boldsymbol{\tau}}_B^3)\right). \tag{3.63}$$

The impedance shaping method implemented prioritization by including the optimized stiffnesses $\boldsymbol{K}_p^{2'}$ and $\boldsymbol{K}_\epsilon^{3'}$ to calculate the control torques and the damping matrix. The damping design method of sec. 3.4.3 automatically detected that no additional joint damping was needed (eqs. (3.38)-(3.39)). The final control torque for the impedance shaping method was:

$$\boldsymbol{\tau}' = \boldsymbol{\tau}^1 + \boldsymbol{\tau}^{2'} + \boldsymbol{\tau}^{3'} - \boldsymbol{\tau}_{B_{\boldsymbol{q}}}{}'. \tag{3.64}$$

For task 2, again, a point on the elbow position was chosen (fig. 3.14). As in experiment 1, a small CTM acted along the yellow and red principal axes of $\mathbf{\Lambda}_p^{2\mathrm{c}}(\boldsymbol{q})^{-1}$. The blue principal axis pointed along a singular direction. The stiffness about this direction was therefore scaled down. The scaling factor for the other two directions was $\gamma = 1$.



(a) CTI of orientation task

(b) Scaling function μ

Figure 3.15: CTI (a) and scaling function (b) for the rotational task 3. The plot colors are consistent with the colors of the coordinate frames in fig. 3.14. The yellow axis showed the minor CTI, since the orientation about this direction was always consistent with the main task and task 2. The blue axis showed the major CTI. The stiffness about the red axis was only partly consistent with the main task and task 2. The direction about the minor CTI was not scaled.

For task 3, a point on the end-effector body was chosen (fig. 3.14). Fig. 3.15a shows the CTI values for task 3 about the principal axes of $\mathbf{\Lambda}_\epsilon^{3\mathrm{c}}(\boldsymbol{q})^{-1}$. For this experiment, an eccentricity threshold of 1400 was set (eq.(3.29)). In this way, a conflict about the blue principal axis direction was deliberately created. This had consequences for the passivity of the controller, as shown later. With insight gleaned from various experiments, the authors propose an eccentricity threshold of $b > 2$ to users of the presented method. This would be a heuristic threshold to detect an eccentric ellipsoid that represents the rotational task consistency (fig. 3.6). The yellow principal axis showed the minor CTI since the robot could maintain orientation about this axis while moving on the circle. Moreover, for the given ori-

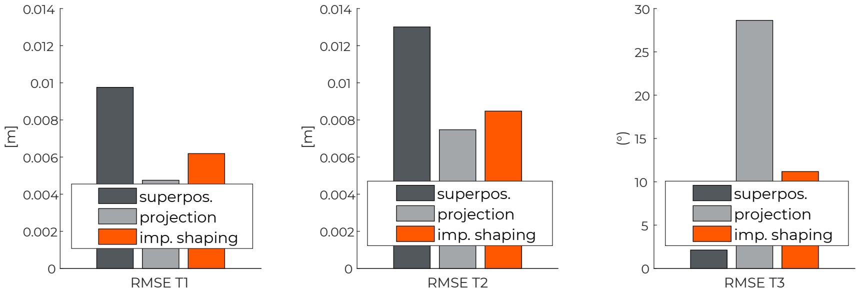


Figure 3.16: Compared to simple impedance superposition, the nullspace projection method and the impedance shaping method reduced the RMSE of the main task (left bar-plot). The nullspace projection method and the impedance shaping method also reduced the RMSE of task 2 (middle bar-plot). Impedance superposition had a higher RMSE for task 2 since task 3 was superimposed. Impedance superposition showed a small error for task 3. Nullspace projection performed worse for task 3 (right bar-plot).

entation about this axis, the elbow position could be maintained. Consequently, the stiffness about this direction was not scaled (fig. 3.15b). While moving around the circle, the CTI about the other two principal axes varied (fig. 3.15a blue and red colors).

Fig. 3.16 shows the RMSE for the main task and both subordinate tasks. Again, the simple impedance superposition showed the largest RMSE for the main task. The nullspace projection method and the impedance shaping method reduced the RSME of the main task by 51.02% and 36.73%, respectively. For task 2, the effectiveness of the nullspace projection method and the impedance shaping method can be seen: They reduced the RSME of task 2 by 42.31% and 34.62%, respectively. To visualize the error of task 3, the rotation matrix between the inertial coordinate frame and the body-fixed end-effector coordinate frame in the equilibrium position was converted to axis-angle representation. The angle was then used for comparison. As expected, simple impedance superposition had the lowest RMSE for task 2. For task 2, the impedance shaping method performed better than the nullspace projection method since the orientation could partly be maintained (fig. 3.15b).

Fig. 3.17 shows the Cartesian errors for experiment 2, again calculated based on the Euclidean 2-norm. For the nullspace projection method minor oscillations still occurred which were a sign of impending instabilities (fig. 3.17, red rectangles).

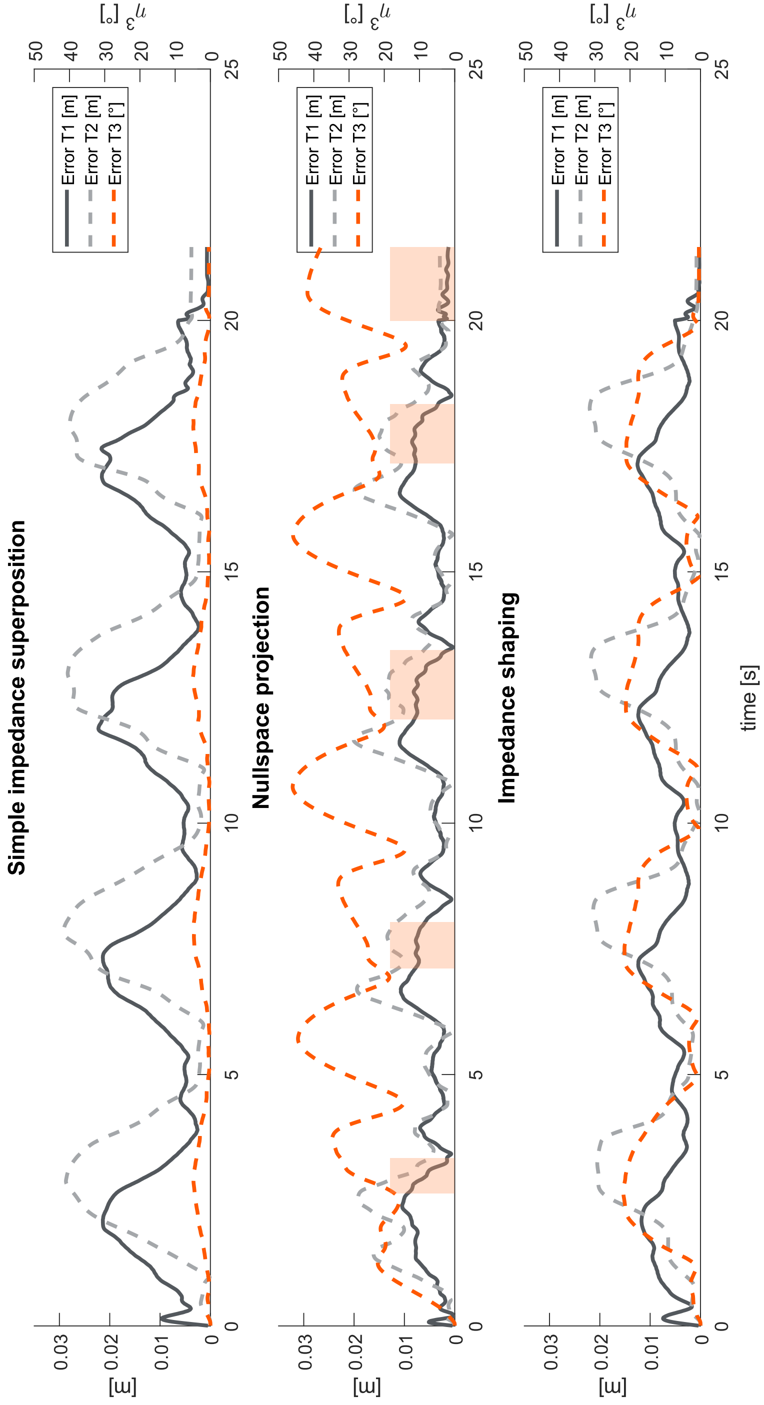


Figure 3.17: Cartesian error for all three controllers. For the nullspace projection method, minor oscillations occurred during the experiment (middle sub-plot, red rectangles).

Due to the choice of the high eccentricity threshold, task 3 injected substantial energy during the experiments with the impedance shaping method. Without the passivity check (extension 3), passivity of the controller was lost (fig. 3.18, upper plot). However, with extension 3, after a task conflict the increase of rotational stiffness was suspended until the dissipated energy was high enough to allow stiffness scaling. In this way, the passivity of the controller was maintained and stability for the robotic system could be guaranteed (fig. 3.18, lower plot). Task 2 was not affected and hence the hierarchical passivity check ensured an improved performance of the main task.

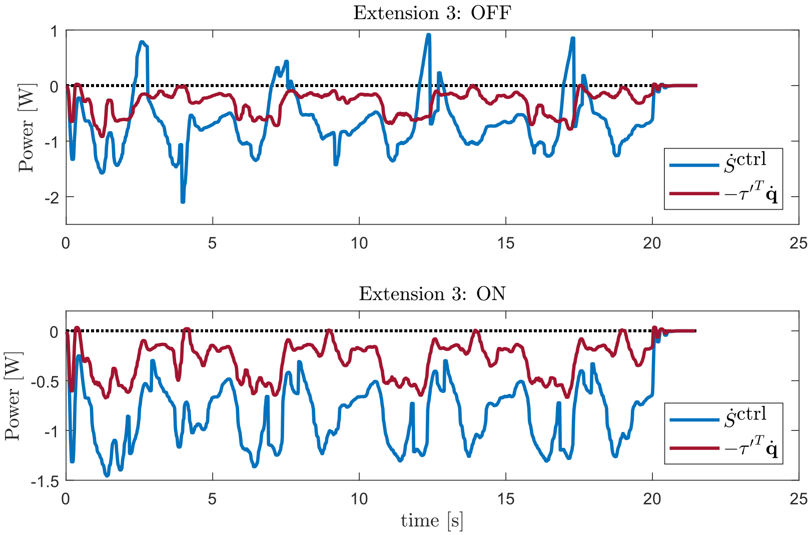


Figure 3.18: Without extension 3, the impedance shaping method violated the passivity of the controller (upper plot). With extension 3, the supplied power was always smaller than the energy flow produced from the time-varying stiffness and hence the controller remained passive during experiment 2 (lower plot). It is hypothesized that the brief epochs of positive power (near 0 seconds and 4 seconds) are due to poor friction compensation during slow joint motions.

Tab. 3.3 shows the minimal, maximal and mean computation time for experiment 2. Again, the nullspace projection method included extension 1 and the novel impedance shaping method included extensions 2 and 3. Compared to the nullspace projection method, the novel impedance shaping method reduced the mean computation time by 74.62%.

3.6 Conclusion

In this chapter, the inertial properties of a robot, expressed in $\mathbf{\Lambda}^{ic}(\boldsymbol{q})^{-1}$, were analyzed to identify if multiple tasks can be performed simultaneously. $\mathbf{\Lambda}^{ic}(\boldsymbol{q})^{-1}$

Table 3.3: Computation time for the controller with 4292 control cycles.

t^{cmp}	**Nullspace projection**	**Impedance shaping**
min	1.10 ms	0.26 ms
max	2.60 ms	1.84 ms
mean	1.30 ms	0.33 ms

was analyzed along its principal directions, which yielded task masses and task inertias, constrained by tasks with higher priority. The key to the impedance shaping method was to transform the task stiffnesses using the principal directions of $\boldsymbol{\Lambda}^{ic}(\boldsymbol{q})^{-1}$. In the aligned directions, the task stiffnesses along/about conflicting directions were scaled. After scaling, the matrices were transformed back to the original coordinates. The scaling method was included in the damping design in order to have a constant damping behavior that was independent of scaling. A great benefit of this general approach is that the impedance shaping method can be applied to all well-established impedance controller implementations that describe the elastic potential based on $\boldsymbol{K}$, e.g., Fasse and Broenink (1997), Caccavale et al. (1999), Stramigioli (2001), and Natale (2003). The key features of this novel control approach are:

1. It is possible to assign a relation between available payload and robot weight for industrial robots. In this work, a relation between virtual constraint masses and robot weight was assigned. By using this relation, a conflict of multiple translational tasks can be detected, independent of the robot type.

2. Constrained task inertias were related to a manipulability ellipoid. The eccentricity of this ellipsoid gave important information about the consistency of multiple rotational tasks. This conflict detection is, again, independent of the robot type.

3. The damping design merges Cartesian damping behavior with joint space damping that does not interfere with the dynamic task behavior. The method automatically detects the dimension of the nullspace and automatically adapts the damping design.

4. The method can be extended to multiple tasks. In this case, a fixed priority is assigned, from a main task with highest priority to subordinate tasks that can be sacrificed if necessary.

5. The control approach is a promising solution for non-expert users, since it auto-tunes the selected task impedances to be consistent with tasks of higher priority.

6. The controller is a promising solution for applications that involve pHRI, since it guarantees a passive robot behavior. Compared to nullspace projection approaches, the method is less dependent on choosing appropriate impedance parameters to achieve a stable robot behavior.

7. The controller extension parallelizes the calculation of kinematic and dynamic data and the optimization of impedance. This has computational advantages and is especially beneficial for real-time control of robots with many degrees of freedom.

3.6.1 Limitations and further work

A limitation of the presented approach is its need for kinematic and dynamic data of the robot. Without access to $\boldsymbol{J}(\boldsymbol{q})$ and $\boldsymbol{M}(\boldsymbol{q})$ of the robot, the impedance shaping approach of sec. 3.4 cannot be performed. However, multiple methods exist to calculate the kinematic data and identify the dynamic robot parameters (Siciliano and Khatib, 2008).

This work assigned multiple tasks to the robot. However, a robot can have additional physical constraints, e.g., obstacles or safe interaction. These constraints can again have conflicts between each other. The prioritization of the main task and the individual constraints depends on the application. In a medical application, for example, the main task that controls the tool behavior usually has the highest priority. For a service robot, compliant behavior in contact might be the highest priority and the main task can be sacrificed if necessary. A promising solution to handle multiple constraints can be found in Osorio et al. (2019).

A robot with many degrees of freedom usually has a large nullspace. If small joint impedances are assigned to the nullspace, the joint motion is not predictable and can lead to joint limit violations. Therefore, it is beneficial to incorporate joint limits as hard constraints, e.g., with the method presented in Hjorth et al. (2021).

The presented approach is an extension for a simple Cartesian impedance controller. Joint impedances are not included since they usually have a negative influence on the desired Cartesian behavior. There might be situations, however, where it is desired to control the postural behavior of the robot (Sentis and Khatib, 2006). In that case, the method of Verdi (2017) can be used and included in the damping design method.

For simple impedance superposition in both experiments, a substantial disruption of the main task by the elbow task could be seen. Compared to this work, the authors in Hermus et al. (2022) superimposed different sets of joint stiffnesses to achieve a predictable joint motion. In both works, the robot had a three-dimensional translational motion task with comparable impedance values (1500 $\frac{kg}{s^2}$ and 1800/2000 $\frac{kg}{s^2}$, respectively). This left a four-dimensional nullspace for additional tasks. Remarkably, Hermus et al. (2022) could show that during forceful interaction, small joint stiffness values yielded an error for the main task that was comparable with nullspace projection. Further work with the controller of this chapter could investigate the effect of a meaningful reduction of the subordinate elbow stiffness, both for a one-dimensional and a four-dimensional nullspace. It could be hypothesized that for a four-dimensional nullspace (experiment 1 with circular motion as main task), simple impedance superposition can yield better results for the main task by an appropriate choice of small task impedances (elbow task) and joint impedances (damping task). For a one-dimensional nullspace, however, it can be assumed that the method presented in this chapter is necessary to improve the performance of the main task.

3.6.2 Parallel control with threading

To control robots with many DOF, usually high-priced controllers with high computational power are needed. To be able to control such robots with low-cost controllers, it is necessary to develop algorithms with low computational effort. The method presented in this chapter proposes an algorithm that increases performance while reducing computation time. This algorithm can, for example, be implemented on a multi-core micro-controller; at present many low-cost multi-core micro-controllers are available.

In this chapter, the calculation time of nullspace projection method was compared with the novel impedance shaping method. The parallel computation through the impedance shaping method reduced computation time significantly (tabs. 3.2 and 3.3). For non-dynamic tasks, however, the parallel computation could also be applied for nullspace projection approaches, e.g., by computing the nullspace projector in a parallel thread.

3.6.3 Selecting the right thresholds for conflict detection

For the detection of a translational task conflict, a relation between constrained task masses and robot weight was assigned. In both experiments, a reasonably small mass ratio ($a = 0.3$) for conflict detection was chosen. However, a high mass ratio would also be feasible, since the constrained task masses along singularities or during task conflicts tend to infinity (fig. 3.10).

For the detection of rotational task conflicts, a relation to manipulability ellipsoids (chapter 2) was established. In experiment 2, a high eccentricity threshold ($b = 1400$) was chosen to deliberately violate the passivity condition and show the effectiveness of controller extension 3 (Passivity check, fig. 3.9). Also, with a small threshold, the performance of the main task would be improved, since rotational stiffnesses along the conflicting directions are scaled down (fig. 3.15).

3.7 Discussion: Similarities to human-motor-control

For constrained tasks, humans rely on mechanical impedance rather than adapting the trajectory or the contact force (Bennett et al., 1992). Moreover, it can be hypothesized that humans also assign priorities to different tasks. Imagine a waiter carrying a full tray. With the same arm, he can position and stiffen his elbow to open a door. However, he will let go of the door if the tray threatens to drop.

Human motion planning is encoded in the central nervous system (CNS). Compared to characteristics in today's robot control, the CNS has slow communication ($\approx 100\ \frac{\mathrm{m}}{\mathrm{s}}$ for neurons) and long feedback loop delays (often >200 ms for voluntary correction) (Burdet et al., 2013). Research showed that the high performance could arise from parallel control of movement and impedance (Maurice et al., 2018).

Humans regularly encounter instabilities, e.g., during walking (Lee et al., 2016) or tool use (Rancourt and Hogan, 2001). Impedances embedded in the human limb are used to regain stability (McIntyre et al., 1996; Burdet et al., 2001). For walking, this is an essential feature since the regulation of displacement would be too slow due to the large moment of inertia associated with the body's center of mass (Burdet et al., 2013).

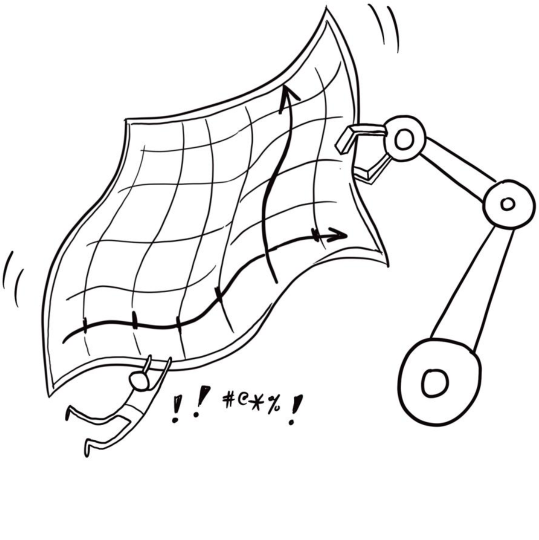
Muscle stiffness is produced by deformation of thick and thin muscle-filaments (Tortora and Derrickson, 2017). The increase of muscle stiffness can then be used for force production (Bennett et al., 1992). Joint viscosity (damping) can be adapted during movement and is—compared to the approach in this chapter—underdamped in general (Perreault et al., 2004). While stiffness and damping can be controlled independent of the posture, they increase monotonically with applied external forces (Huang et al., 2020).

Stiffness and damping are related to inertia: The moment of inertia about a joint contributes to impedance (Burdet et al., 2013). The inertial properties depend on the configuration of the limb and may be optimized for the current tasks (Hogan, 1985c). For example, the hand has low inertia to allow rapid response. Similar to the control approach in this chapter, humans are aware of their inertial properties since postures are optimized to comply with given constraints (Lacquaniti et al., 1993).

The aforementioned relations between human motor control and impedance control were the main inspiration for this work. The impedance shaping method presented in this chapter could improve the task performance of a simple impedance controller, while providing passivity and computational efficiency. However, to reproduce human-like performance, a lot remains to be done. Further work should use the controller presented here as a basis for optimization of the robot impedance behavior during forceful contact tasks.

CHAPTER 4

Energy budgets for coordinate invariant robot control in pHRI



Lachner J, Allmendinger F, Hobert E, Hogan N and Stramigioli S (2021) Energy budgets for coordinate invariant robot control in physical human–robot interaction. *The International Journal of Robotics Research* 40(8-9): 968–985. doi: 10.1177/02783649211011639

Lachner J and Stramigioli S (2021) Method and system for operating a robot. URL `https://patents.google.com/patent/DE102020209866B3`

4.1 Introduction

As mentioned in sec 3.1, there exist a considerable number of collaborative robots (International Federation of Robotics, 2019b). However, despite the prognosis of market researchers, collaborative robot applications are still a minority in the industrial sector. While 400,000 industrial robots were installed in 2018 (International Federation of Robotics, 2019a), only 12,000 professional service robots were instated in the same year[1] (International Federation of Robotics, 2019b). One reason is the extensive and complicated certification process required to integrate a collaborative robot into a manual work process. The process usually begins with a virtual set-up in simulation. First, risk assessments have to be performed to identify possible hazards for the human when entering the robot workspace (International Organization for Standardization, 2011, 2010). The objective of the risk assessment is to define how likely it is that the hazards harm the human and to provide measures to minimize these risks. Next, the real robot is installed and programmed with a range of velocities. After the robot process is implemented, the impact force of the robot in case of a collision is verified with a measurement device (Dombrowski et al., 2018).

The international standard ISO/TS 15066 (International Organization for Standardization, 2016) defines bio-mechanical limits for each body region of the human. Thresholds are defined for two different contact scenarios: transient contact (i.e., collision) and quasi-static contact (i.e., clamping). A collision is a dynamic impact with the moving robot. A clamping scenario occurs when a robot continues to follow a pre-planned trajectory even though an obstacle has been encountered. In the risk assessment, the programmer of the robot application has to decide which contact scenario is likely to occur. Moreover, the affected body regions have to be identified for each part of the robot application. In an iterative manner, the robot velocity is reduced until no bio-mechanical threshold is violated (fig. 4.1).

Typical robot controllers are divided into two successive stages: motion planning and motion execution. During motion planning, a reference trajectory is computed for every robot joint. The result is then passed to the motion execution stage, during which the planned joint trajectories are tracked as accurately as possible (Siciliano et al., 2009). For pHRI, these controllers rely on the prediction of robot collisions with the environment (Phan et al., 2011; Lee and Song, 2015; Bergner et al., 2019). Since the reference trajectory cannot be modified during motion, the robot executes a stop reaction if a safety-related signal exceeds a pre-defined threshold (e.g., the distance between robot and environment is too small or the collision force with the environment is too large). In ISO 10218-1 (International Organization for Standardization, 2011), robot motion is defined as the main risk source for the human. Therefore, only stop reactions are considered to be safe. In constricted workspaces where the robot and human work closely together, this can easily lead to clamping dangers. Moreover, for situations where the robot brakes are activated during a quasi-static contact, the human has no

[1] Excluding autonomously guided vehicles and inspection robots.

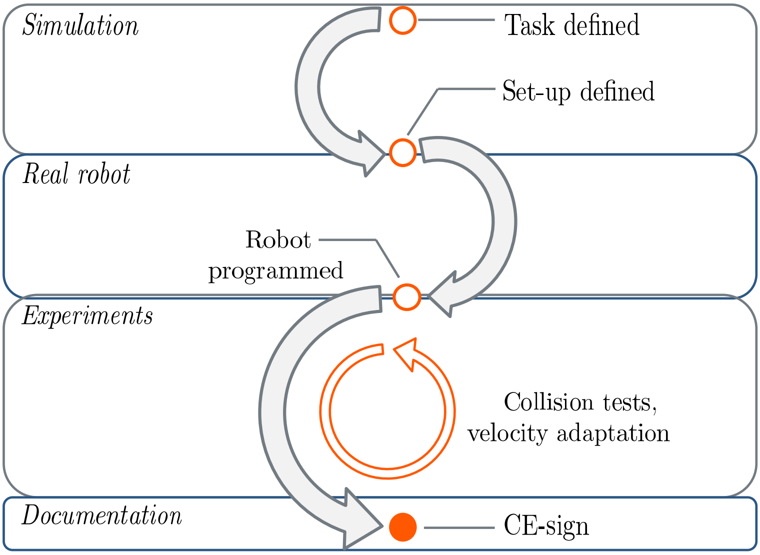


Figure 4.1: Simplified certification process of a collaborative robot cell. The robot movement is roughly planned in simulation. To satisfy the force thresholds in ISO/TS 15066, possible contact scenarios are identified and respective body regions are selected. The robot process is programmed with a range of velocities. Collision measurements are performed in order to evaluate the robot impact. Iteratively, the velocity of the robot is adapted. To prove compliance with existing standards and regulations, extensive documentation is needed.

means to escape from a clamping scenario (De Luca et al., 2006).

Reactive control schemes merge the planning and execution phases. Therefore, it is possible to develop unified strategies for collision detection and reaction (Haddadin et al., 2008a). Several approaches for collision detection have been proposed (Fritzsche et al., 2011; Haddadin et al., 2017; Liang et al., 2020; Birjandi et al., 2020). If the human and robot share the same workspace, the collision point on the robot structure is not known in advance. Therefore, it is important that the detection does not depend on the impact location, nor on the current robot configuration. Different safety-related reaction strategies exist. One approach is to apply monitoring functions during the robot movement and automatically adapt the controller parameters during run-time (Haddadin et al., 2008b; Navarro et al., 2016; Raiola et al., 2018; Muñoz Osorio et al., 2019; Ferraguti et al., 2020). The goal of those controllers is to limit the potential robot impact in case of an unintended collision. Other work focuses on active robot reactions that are executed after a collision has been detected (Laffranchi et al., 2009; De Luca and Flacco, 2012; Khan et al., 2014). For unstructured environments, it is important to ensure that these active reactions do not lead to new risks. Moreover, no controlled robot reaction is fast enough to have a significant influence on the first collision impact (Haddadin et al., 2008c,b).

A promising field of research is learning based approaches to support pHRI, e.g., imitation learning (Billard et al., 2008; Laskey et al., 2016; Manschitz et al., 2020) and dynamic motion primitives (Schaal, 2006; Ijspeert et al., 2013). The goal of these approaches is to demonstrate to the robot human-like interaction and therefore resolve unsafe behavior.

Learning based approaches often depend on appropriate sensors to monitor the environment. Even though considerable research in safety-related perception and vision has been done in recent years (Flacco et al., 2015; Beetz et al., 2015; Flacco and De Luca, 2017; Charalampous et al., 2017; Chen and Song, 2018; Sadrfaridpour and Wang, 2018), there is a significant lack of certified sensors, e.g., to calculate distances, to predict human intentions, and to supervise sharp objects at the robot end-effector. If such sensor functions should be used for pHRI, they have to fulfill the strict requirements of ISO 10218-1 and ISO 13849-1. This means that for a severe and frequently appearing risk, the safety-related software has to be designed in a redundant manner (i.e. "Category 3") and the probability that failures of the safety system remain undetected has to be less than 10^{-6} (i.e. "Performance Level d" or "System Integrity Level 2") (International Organization for Standardization, 2011, 2015). Since these sensors are largely unavailable, ISO/TS 15066 excludes the use of sharp objects in a pHRI application.

To program collaborative applications, coordinate frames are typically placed on the robot structure, e.g., to monitor Cartesian velocities or predict external forces. As may be anticipated, unmonitored body parts on which no coordinate frame has been placed can easily cause risks for the human. Imagine a service robot working in a supermarket surrounded by children. To be productive, it is desired that such a robot moves at high velocity. However, the maximum velocity of the robot has many constraints. Some of the constraints are robot-related (e.g., movement direction and curvature) and others are due to the given environment (e.g., clamping dangers and number of surrounding people). Therefore, it is hard to define possible collision points and maximum limits for the robot velocity. For such applications, intrinsically safe controllers are needed that automatically adapt the robot behavior and exclude clamping danger.

Physical interaction can be characterized by energy exchange (Colgate and Hogan, 1988). A major advantage is that energy is a coordinate invariant entity. Energy shaping techniques in robotic control have already been used in 1981 (Takegaki and Arimoto, 1981). However, there is only limited work addressing the control of energy for safe pHRI. In most energy-related control approaches, the energy difference between two successive control cycles is compared in order to make a statement about the passivity of the robot. An introduction to passivity was given in sec. 3.2.1. Probably the most prominent passivity concept is the "energy tank" and "energy routing" approach originally proposed by Duindam and Stramigioli (2004). This approach can be used for telemanipulation (Franken et al., 2011; Ferraguti et al., 2013, 2015), to guarantee stability during contact (Schindlbeck and Haddadin, 2015; Shahriari et al., 2017, 2020), or to render nullspace projection approaches passive (Dietrich et al., 2016, 2017; Garofalo and Ott, 2018). Moreover, the energy tank concept has been proposed for safe pHRI (Tadele et al., 2014; Raiola et al., 2018). An energy tank constrains the robot energy to a finite amount. During movement, the robot extracts energy from the tank. In all approaches, the initial amount of energy in the tank is crucial. For pHRI, this can lead to a conflict: On the one hand, the available energy should be high enough for the robot to fulfill its task; on the other hand, allowing a high energy value can pose a risk for the human in case of an unintended collision. Therefore, it might be worth exploring the allocation of an *energy budget* for each control cycle (Groothuis et al., 2018).

Chapters 2 and 3 established methods for the analysis and control of collaborative robots. This chapter focuses on the use of collaborative robots in pHRI applications. The coordinate dependency of the implementation of safety measures in pHRI applications is addressed. The current certification process is reviewed with the help of an example application on a real robot. For the certification process, coordinate frames are used to monitor Cartesian velocities and predict external forces. It will be shown that for a given robot task these strategies do not generally predict unsafe robot behavior. The novelty of this chapter is to assign a dynamic relationship to a contact scenario. This means that the contact is not only defined by the applied robot force but also influenced by the interacting environment. The control approach of this chapter assigns an energy budget to the robot to restrict the amount of energy flow in case of a collision. The robot is exposed to an artificial potential field and auto-tunes this potential field at runtime to ensure that the energy budget is not violated. Summarized, the main contributions of this chapter are:

- Illustration of the current certification process of applications with pHRI on the real robot with state of the art certification equipment.
- Identification of causal energy flow during contact scenario.
- Allocation of safe energy budget to guarantee safe robot behaviors during collisions and clamping scenarios.
- Significant reduction of the programming complexity of pHRI applications.
- Experimental validation of the novel approach with state of the art certification equipment.

The results of this chapter are partly reported in Lachner et al. (2021).

4.2 Safe pHRI: Classical approach

In order to certify a robot application, a risk assessment based on ISO 12100 has to be performed. The first step of the risk assessment is to list possible risks for the human. For pHRI, two major hazards are collisions and clamping scenarios, which will be treated in the remainder of this chapter.

4.2.1 Certification process of pHRI cells

Each robot application can roughly be divided into transfer motion, approach motion, and process. During the transfer motion, collisions or clamping scenarios can occur. The approach motion mostly yields clamping dangers. In fig. 4.2, an exemplary robot application is shown which will be used in the following to demonstrate the current certification process. If a pHRI application is placed in a restricted and unstructured workspace, the contact scenario cannot be determined in advance and the programmer of the robot cell has to assume the worst case scenario. To determine the impact of a collision between the robot and the human, possibly affected body parts have to be identified. ISO/TS 15066 provides bio-mechanical thresholds for different body parts of the human. Moreover, the stiffness k $\in \mathbb{R}$ of each body part is listed. Note that ISO/TS 15066 permits contact with face, skull, and forehead. If it is likely that those areas are affected, equipment has to be provided to protect the human co-worker. For the example application (fig. 4.2), the upper arm, lower arm, and hand were identified as the affected body parts. The respective bio-mechanical thresholds and stiffness values can be seen in tab. 4.1[2]. Note that the transient force and pressure thresholds are twice as high as the respective quasi-static values.

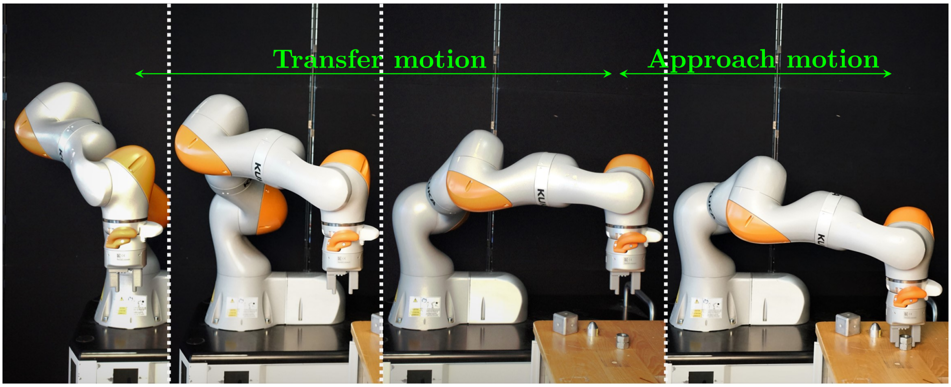


Figure 4.2: A typical robot application is divided into transfer motion and approach motion. At the end of the transfer motion, the robot is pre-positioned. Relative to this position, the workpiece is approached.

In the next step of the risk assessment, possible collision points on the robot structure have to be determined in order to calculate the impact surface area during contact. The robot programmer analyzes the robot motion and selects the sharpest points on the robot structure, i.e., the surface with the smallest curvature. Often, the most exposed surface in the movement direction is selected. For the example application, the white rounded cover around the cables of the robot flange was selected as the most exposed surface that is likely to collide with the human arm during transfer motion (fig. 4.2). During the approach motion, it is possible

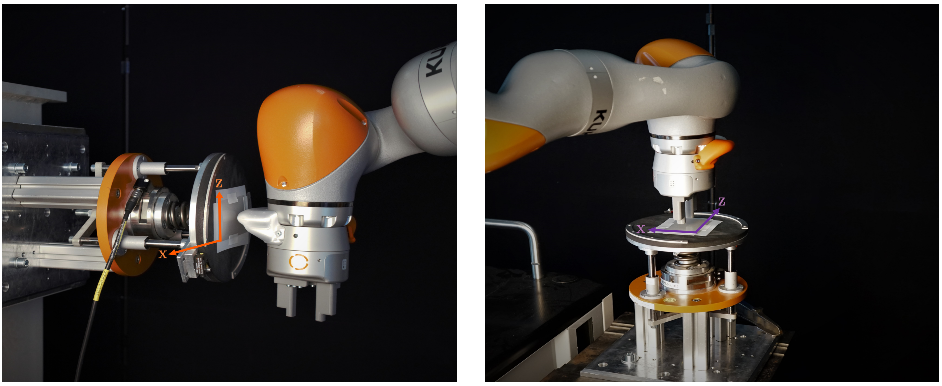
[2] To facilitate comparison, the units of ISO/TS 15066 will be used.

Table 4.1: Bio-mechanical thresholds, based on ISO/TS 15066, for affected body parts in the example application. The bold numbers indicate the lowest values that have to be taken into account.

	Transient		**Quasi-static**		
	F[N]	P[N/cm^2]	F[N]	P[N/cm^2]	k[N/mm]
Up. arm	300	380	150	190	30
Lo. arm	320	**360**	160	**180**	40
Hand	**280**	380	**140**	190	75

that the human arm or hand becomes clamped between the table and the gripper jaws, which have a flat surface at the finger tips (fig. 4.2).

To analyze the severity of a collision and to guarantee conformity with ISO/TS 15066, each pHRI application is certified based on measurements. The desired output of the measurements are impact surface area and collision force. With this information, contact pressure can be calculated. For the example application, such measurements were conducted. In fig. 4.3, the measurement setup for the transfer motion and the approach motion can be seen, with detailed technical specifications presented in appendix G.1. Two test series were performed, one for the transfer motion and one for the approach motion.

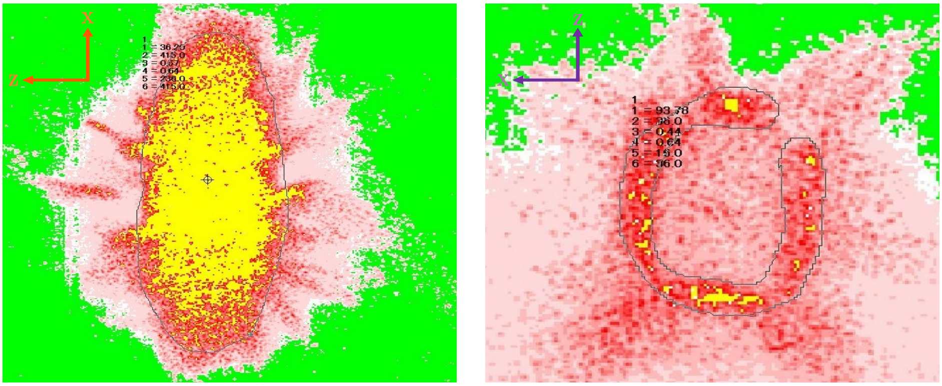


(a) Collision test with gripper cover during transfer motion

(b) Collision test with gripper jaws during approach motion

Figure 4.3: To determine the impact surface area and impact force during transfer motion (a) and approach motion (b), a collision measurement device was mounted on solid pillars. The robot application was started and the identified robot part collided with the measurement device. The tests were conducted with varying robot speeds to determine the maximum admissible bio-mechanical threshold.

The collision measurements were conducted with a measurement device that was mounted on a solid steel pillar. This simulated a quasi-static contact in which the human has no possibility to back up. The measurement device was equipped with a force sensor. On top of the force sensor, a spring was attached that simulated the stiffness of the human body part. The robot collided with a foam rubber pad that resembled the stiffness of the skin. In the experiments, the spring constants used were 25 N/mm (arm) and 75 N/mm (hand), for transfer motion and approach motion, respectively. These values emulated the stiffness values of the affected body parts, defined in tab. 4.1. The foam rubber had a thickness of 0.01 m. To assess the impact surface area, a pressure sheet was fixed on the measurement device during the first test of each series (fig. 4.3). Using the software of the pressure sheet provider, the impact surface area was determined by outlining the area manually in the software (thin black line in fig. 4.4). For the experiments, the impact surface area for the transfer motion was $A_t = 4.15 \text{ cm}^2$ (gripper cover) and for the approach motion $A_a = 0.36 \text{ cm}^2$ (gripper jaw). As a last step of the risk assessment, safety measures have to be provided to keep the collision impact within an acceptable range. For contact scenarios, this means that the bio-mechanical thresholds of ISO/TS 15066 must not be exceeded. Collaborative robots provide measures to lower the impact during a contact scenario. For the KUKA LBR iiwa, integrated torque sensors can be used to detect a collision. Moreover, based on the torque sensors, external Cartesian forces can be predicted. In the experiments, a collision detection criterion of torque > 20 Nm was set. If any torque sensor measured a value above this threshold, a stop reaction was triggered and the robot brakes were activated. This stop reaction was implemented because ISO 10218-1 argues that the moving robot is a risk and hence only a stopped robot is safe. At the end of the collision measurements, the maximum robot speed can be determined such that the collision detection acts quickly enough to stay within



(a) Impact surface area for transfer motion: $A_t = 4.15 \text{ cm}^2$

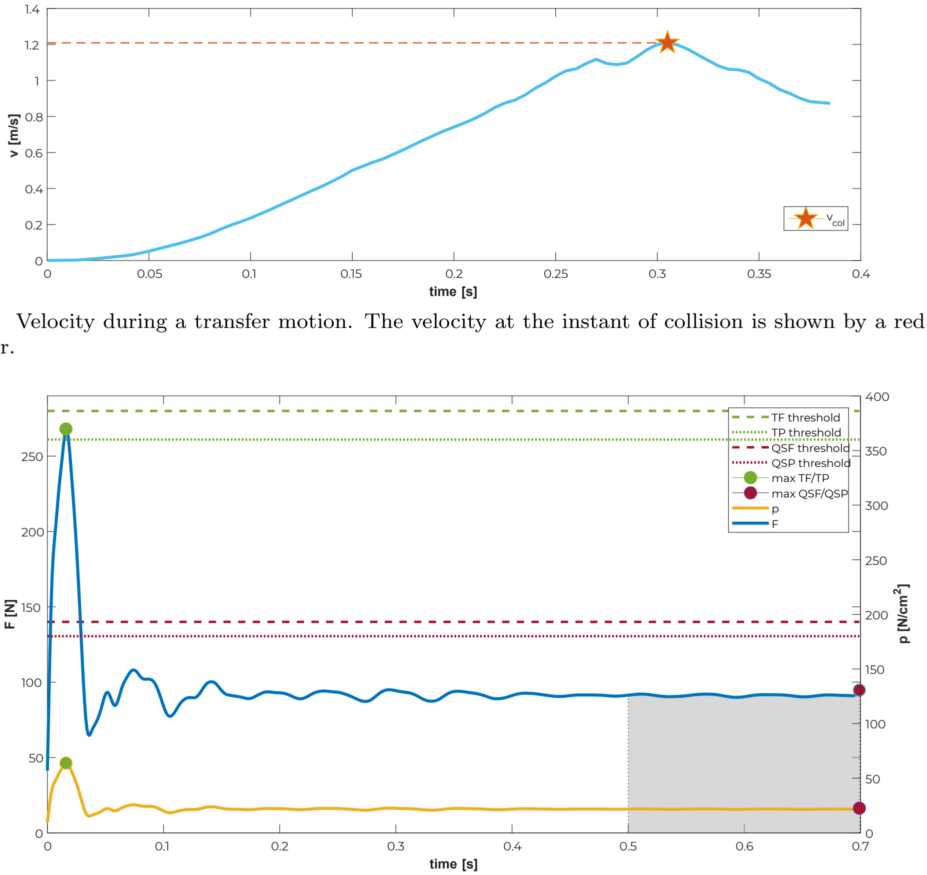
(b) Impact surface area for approach motion: $A_a = 0.36 \text{ cm}^2$

Figure 4.4: After scanning the pressure sheet, the respective pressure regions are displayed in different colors. Yellow represents the area with highest contact pressure. Red indicates the second highest contact pressure, from high in dark red to low in light red. Green represents area without contact. A contour was manually drawn to determine the impact area. The impact area of the gripper cover and one gripper jaw are visualized in (a) and (b), respectively.

the bio-mechanical limits.

Every contact scenario has a transient and a quasi-static phase. During each phase, maximum force and pressure values are tracked. While the maximum values are considrered over the complete impact time for transient contact, the maximum quasi-static values are assumed to be established 0.5 seconds after first contact (International Organization for Standardization, 2016). For the example application, an unstructured environment and a quasi-static contact scenario for the transfer and approach motions were considered.

Fig. 4.5a shows the robot end-effector velocity in the direction of movement during the transfer motion. Approximately 0.3 s after the robot motion started, a collision with the measurement device occurred. The velocity at the first impact instant (v_{col}) was analyzed by detecting large torque deviations in axis 6 of the robot. In fig. 4.5b, the respective force recordings of the measurement device can be seen. The recording was automatically started after a force threshold of > 20 N was exceeded. Since the cover of the gripper was rounded and the contact surface area was relatively large (fig. 4.4a), low transient pressure values and low quasi-static pressure (QSP) values were observed. The maximum transient force value was close to the bio-mechanical force threshold of the hand (cf. tab. 4.1 and fig. 4.5b). Hence, v_{col} was close to the maximum robot velocity that remained within the allowable bio-mechanical thresholds if a collision occurred during the transfer motion. Collaborative robots provide safety-related monitoring functions that comply with ISO 13849-1 (Ferraguti et al., 2020). For the KUKA LBR iiwa, one such function can monitor the Cartesian velocity in a given task coordinate frame. Consequently, v_{col} would have to be set as safety-monitored velocity. If this velocity is exceeded, a stop reaction is triggered.

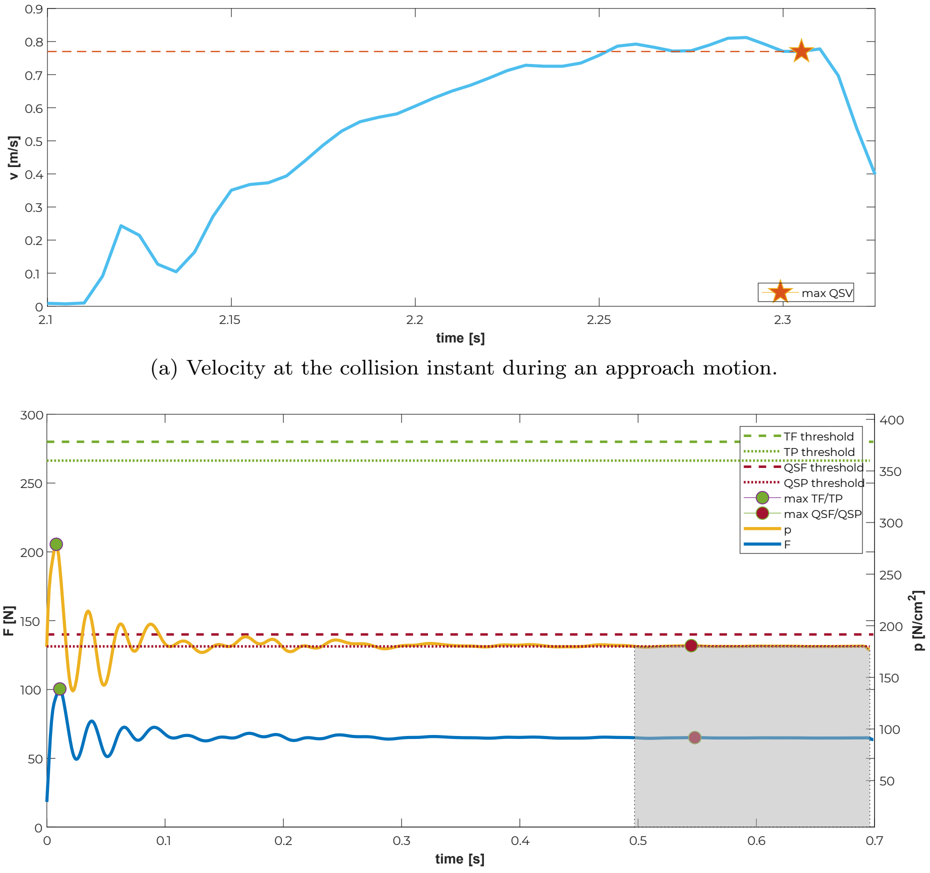


(a) Velocity during a transfer motion. The velocity at the instant of collision is shown by a red star.

(b) Force and pressure during quasi-static contact.

Figure 4.5: (a) For each trial, the velocity at the first impact instant was determined. (b) Force and pressure distribution for a collision during the transfer motion. While the maximal transient pressure and maximal quasi-static pressure were far lower than the admissible threshold, the transient force was close to the permitted threshold of 280 N. Even though the robot moved at high velocity (fig. 4.5a), the stop reaction reduced the force peak in a short time (cf. F at 0.016 s $\leq t \leq$ 0.036 s). Once the brakes were activated, a quasi-static force of $\leq$ 94.74 N and quasi-static pressure of $\leq$ 10.13 N were maintained (red dots).

Fig. 4.6a shows the robot end-effector velocity in the movement direction of the approach motion. Unlike the transfer motion, high QSP values were observed since the gripper jaws had a small contact surface area (cf. fig. 4.4b and fig. 4.6b). The robot velocity was iteratively adapted, until the QSP approximated the threshold for the lower arm (cf. tab. 4.1 and fig. 4.6b). Note that v_{col} does not necessarily represent peak velocity since velocity oscillations due to the impedance controller can occur (cf. fig. 4.6a between 2.25 s and 2.3 s). In the example application, transient force values were critical for collisions with round robots parts (e.g., the gripper cover), while QSP values had to be considered for parts with small impact surface areas (e.g., the gripper jaws).



(a) Velocity at the collision instant during an approach motion.

(b) Force and pressure during quasi-static contact.

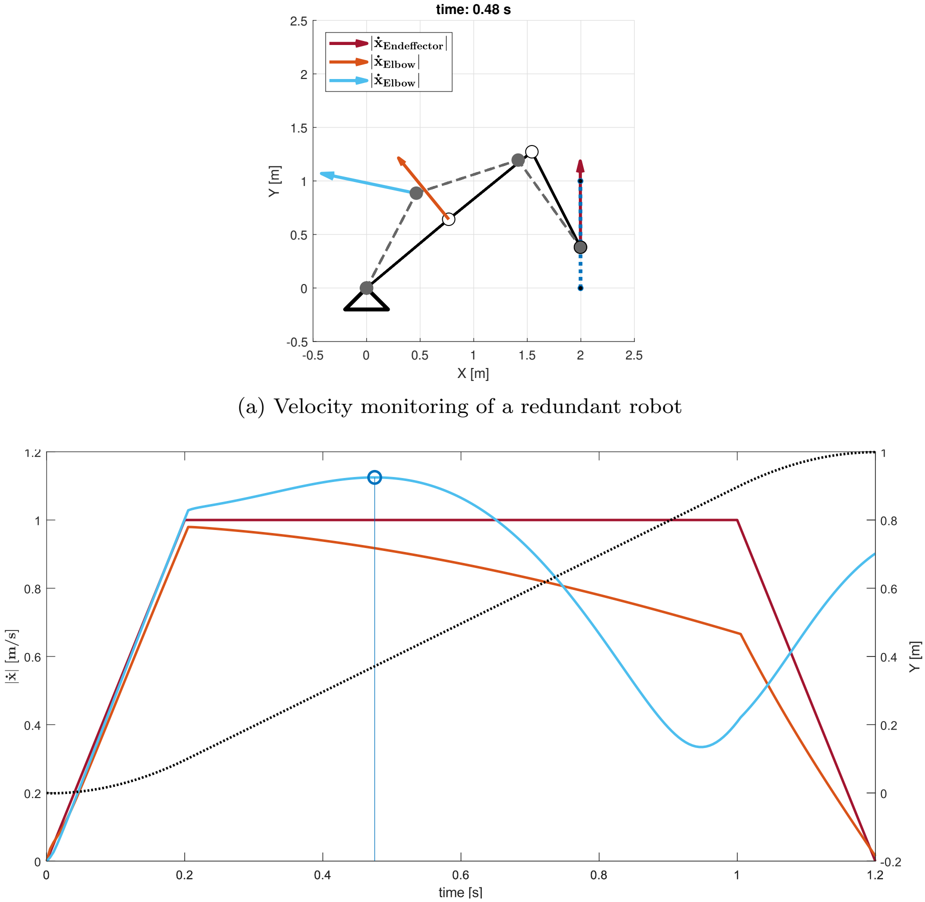
Figure 4.6: The robot velocity was iteratively adapted until the contact pressure value met the bio-mechanical threshold of $180\frac{\mathrm{N}}{\mathrm{cm}^2}$. All other force and pressure values were far lower than the allowed thresholds (tab. 4.1).

4.2.2 Specific aspects of the safety implementation

As shown in the previous section, the robot programmer has to determine critical points on the robot structure. For these points, coordinate frames are placed and the maximal velocity is evaluated. As already mentioned in sec. 2.2.1, different Jacobian matrices exist. Most commonly, the Hybrid Jacobian Matrix (sec. 2.2) $\boldsymbol{J(q)}_{\mathrm{H}} \in \mathbb{R}^{6 \times n}$ is used, which maps joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ to linear and angular

Cartesian velocities $\boldsymbol{V} \in \mathbb{R}^6$ (eq. (2.6)). The matrix $\boldsymbol{J(q)}_{\mathrm{H}}$ is expressed in the body-fixed coordinate frame and expressed with respect to the inertial coordinate frame.

To analyze a scalar value, the vector $\boldsymbol{V}$ is transformed to a scalar velocity $v \in \mathbb{R}$, usually by using the Euclidean 2-norm. In the kinematically redundant case, there are infinitely many solutions $\dot{\boldsymbol{q}}$ to produce a desired Cartesian velocity. Hence, for a given velocity $\boldsymbol{V}$ the joint velocity of the robot is not unique (eqs. (2.9), (2.10)). This can lead to high joint velocities which can be a risk for the human. This raises the question where to put the coordinate frame on the robot structure. For a poor choice, unmonitored high velocities can cause injuries in case of a collision (fig. 4.7).



(a) Velocity monitoring of a redundant robot

(b) Cartesian velocity magnitudes based on a Euclidean 2-norm

Figure 4.7: A redundant 3-DOF robot moving along the y-direction (dotted line in fig. 4.7b). The given task motion can be produced by infinitely many joint velocities. Two different joint motions with different elbow velocities are shown. The arrow colors in fig. 4.7a correspond to the graph colors in fig. 4.7b. All velocities are calculated based on the Euclidean 2-norm. In the given time frame (vertical blue line in fig. 4.7b), the elbow velocity (blue color) is higher than the velocity of the end-effector (dark red color). If this velocity is not evaluated during the certification process, it can represent a risk for the human.

To lower the impact of a collision, often joint torque sensors are used to detect a contact with the environment. External torques $\boldsymbol{\tau}^{\text{ext}} \in \mathbb{R}^n$ are estimated based on the difference between the measured torques, control torques, and acceleration estimations and hence require accurate model data. However, $\boldsymbol{\tau}^{\text{ext}}$ cannot always be used as a safety measure. For assembly processes, the robot cannot differentiate between an intended or unintended contact. Another commonly used approach is to calculate external forces $\boldsymbol{F}^{\text{ext}} \in se^{\star}(3)$ acting on the robot structure. For robots with joint torque sensors, $\boldsymbol{F}^{\mathrm{ext}}$ can be obtained by using the Generalized Inverse Jacobian (eq. (2.10)) $\boldsymbol{J(q)}_{\mathrm{H}}^{\#} \in \mathbb{R}^{n \times 6}$:

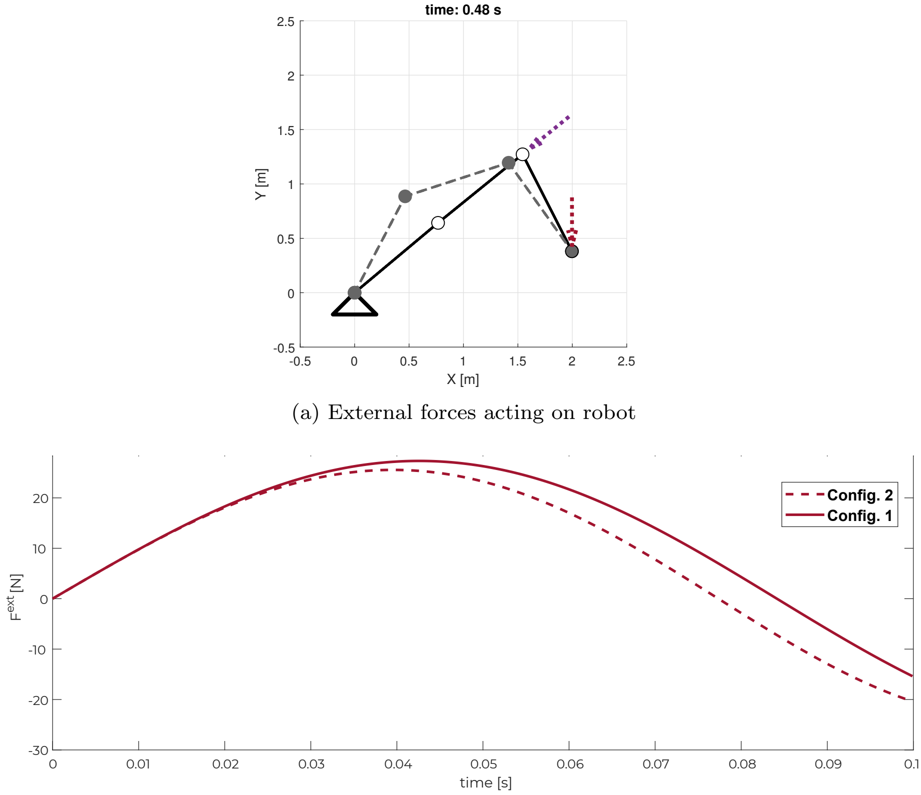
$$\boldsymbol{F}^{\mathrm{ext}} = \left(\boldsymbol{J(q)}_{\mathrm{H}}^{\#}\right)^{T} \boldsymbol{\tau}^{\mathrm{ext}}. \tag{4.1}$$

For a kinematically redundant robot, $\boldsymbol{J(q)}_{\mathrm{H}}^{\#}$ is not unique and commonly depends on the choice for a cost function and a weighting matrix (sec. 2.2.1). $\boldsymbol{J(q)}_{\mathrm{H}}^{\#}$ also depends on the coordinate frame on the robot structure. Moreover, the measurement of $\boldsymbol{\tau}^{\mathrm{ext}}$ critically depends on the current robot configuration as well as on the position and direction of the applied force. Along singular directions, no forces can be detected (fig. 4.8a).

The contact force is frequently described by a linear mass-spring-mass model (Haddadin et al., 2009; International Organization for Standardization, 2016; Rosenstrauch and Kruger, 2017). For a given direction, represented by the unit vector $\boldsymbol{u} \in \mathbb{R}^{3}$, the magnitude of the scalar force due to the linear mass-spring-mass model depends on the robot's "reflected inertia" (Haddadin et al., 2008b) $m_{\mathrm{eff}} \in \mathbb{R}$ at the contact point (eq. (2.27)) (Wassink and Stramigioli, 2007; Haddadin et al., 2010; Khatib, 1995):

$$m_{\mathrm{eff}} = \left(\boldsymbol{u}^{T} \boldsymbol{\Lambda}^{-1} \boldsymbol{u}\right)^{-1}. \tag{4.2}$$

The "mobility end-point tensor" (Hogan, 1984; Lachner et al., 2020) $\boldsymbol{\Lambda}^{-1} \in \mathbb{R}^{6 \times 6}$ (eq. (2.18)), derived in sec. 2.3, is a static measure and solely depends on the robot configuration. Hence, if the contact force is modeled based on $\boldsymbol{\Lambda}^{-1}$, different nullspace motions will cause different predictions of contact forces (fig. 4.8b).

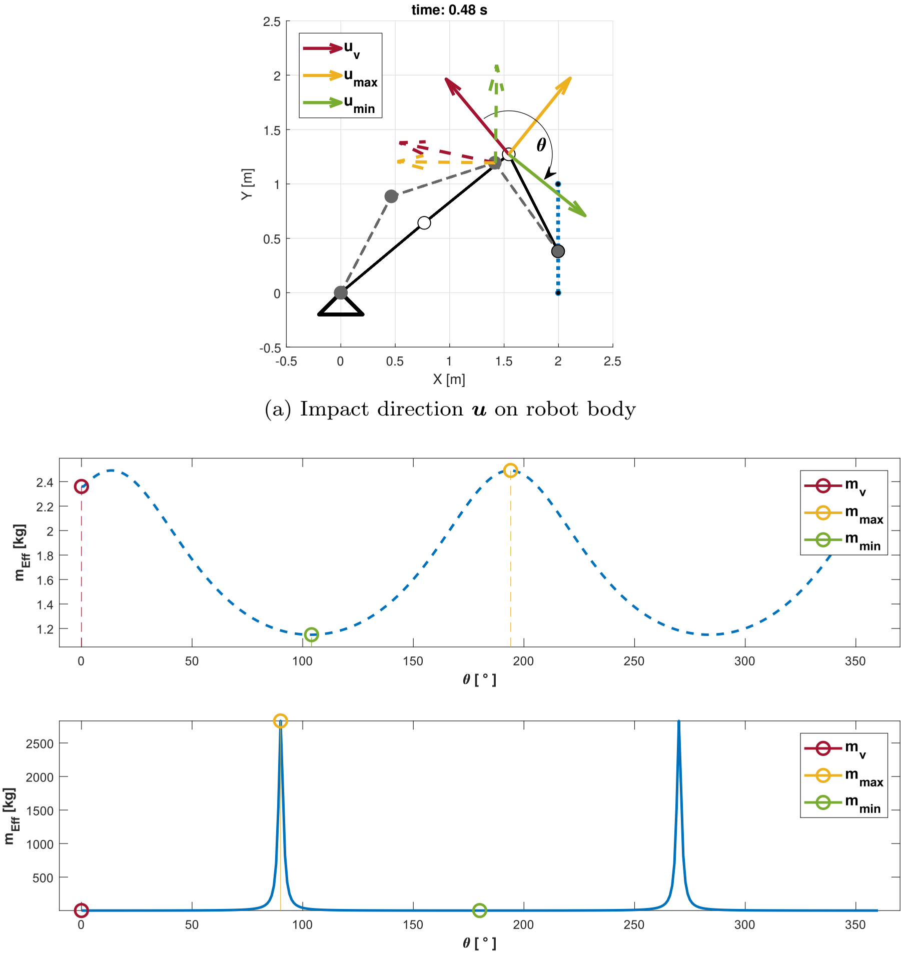


(a) External forces acting on robot

(b) Linear force based on mass-spring-mass model

Figure 4.8: (a) While the end-effector is moving upwards along the y-axis, two external forces (purple and red arrow) act on the robot. If the calculation of $\boldsymbol{F}^{\text{ext}}$ is based on joint torque sensors, the purple force cannot be measured for robot configuration 1. (b) Applying the linear mass-spring-mass model in Haddadin et al. (2009), the simulated contact force (red color) yields two different solutions, depending on the robot nullspace motion.

As can be seen in eq. (4.2), the magnitude of m_{eff} depends on the direction of the unit vector $\boldsymbol{u}$. For the end-effector body, the direction along the linear end-effector motion can be monitored. However, when selecting other points on the robot structure, different nullspace motions result in different values of m_{eff} (fig. 4.9). Usually, the normal direction to a point on the impact surface with the smallest curvature is selected (Wassink and Stramigioli, 2007; Haddadin et al., 2008b). Near singular configurations, the magnitude of m_{eff} is highly sensitive to small direction changes (fig. 4.9, bottom plot on right side). For a kinematically redundant robot with an almost round impact surface along the robot structure, it is difficult to determine an appropriate impact direction.



(a) Impact direction $\boldsymbol{u}$ on robot body

(b) Magnitude of m_{eff} along u

Figure 4.9: Reflected inertia for two different feasible joint space trajectories with coordinate frames on joint 3 (fig. 4.9a). The magnitude of m_{eff} for all unit directions $\boldsymbol{u}$ can be seen in fig. 4.9b. Rotating the coordinate frame on the selected body yields different solutions of m_{eff}. The yellow direction $\boldsymbol{u}_{\text{max}}$ shows the maximal value of m_{eff}. The green direction $\boldsymbol{u}_{\text{min}}$, with minimal value of m_{eff}, is perpendicular to $\boldsymbol{u}_{\text{max}}$. The reflected inertia along the current linear motion $\boldsymbol{u}_v$ for the selected point is displayed in red. For two different nullspace motions, different magnitudes of m_{eff} arise. Near singular configurations, a small direction change leads to a high change in magnitude of m_{eff} (fig. 4.9b, lower plot).

In the example application, one coordinate frame was placed on the end-effector to monitor and adapt its velocity. Additional high velocities, e.g., of the robot elbow, were not tracked and can represent an injury risk. Even if all appropriate coordinate frames have been placed, the clamping danger still exists. To comply with ISO 10218-1, the robot stops and the brakes are activated if an unintended contact is detected. This is especially disadvantageous for applications with restricted workspaces. This can be seen in the example application (fig. 4.5b and fig. 4.6b): The remaining quasi-static force (QSF) values after the robot stopped were approximately 94.74 N and 65.06 N for the transfer motion and the approach motion, respectively. For the approach motion, a constant pressure of 180 N/cm^2 remained. For the example application, the force and pressure values are in correspondence with ISO/TS 15066. However, a wrong choice of considered body parts (tab. 4.1) will not eliminate all risks. Even if all affected body parts were identified correctly and all contact values were in correspondence with ISO/TS 15066, the programmer has to provide some means to release the human co-worker.

4.3 Coordinate invariant control of robot energy

A collision between a robot and its environment is a dynamic interaction and hence cannot be described with either velocities or forces alone (Stramigioli, 2015; Folkertsma and Stramigioli, 2015). A physical contact results in a robot response. It is important that this reaction should not evoke new dangers for the human (Haddadin et al., 2017). In the presented approach, the energy of the robot will be controlled, which is a coordinate invariant measure and therefore does not depend on the selection of coordinate frames. The controller auto-tunes the total energy of the robotic system. The compliant robot behavior in quasi-static contact will protect the human from clamping scenarios.

In the remainder of this chapter, all entities are expressed in the body-fixed coordinate frame and the Geometrical Body Jacobian of eq. (3.2) is used. To make the notation light it will be defined: $\boldsymbol{J}(\boldsymbol{q}) := \boldsymbol{J}^{\mathrm{b}}(\boldsymbol{q})_{\mathrm{G}}$ and $\boldsymbol{\xi} := \boldsymbol{\xi}^{\mathrm{b}}$.

4.3.1 Control of potential energy

Due to the virtual spring of the impedance controller, the robot is exposed to a potential energy function (Stramigioli (2001), sec. 3.2). The potential energy can also be a function of the robot joint displacement $\Delta\boldsymbol{q} = (\boldsymbol{q}^{\mathrm{e}} - \boldsymbol{q}) \in \mathbb{R}^n$ and the joint stiffness $\boldsymbol{K}_q \in \mathbb{R}^{3\times3}$:

$$\mathscr{U}(\Delta\boldsymbol{q}) = \frac{1}{2}\Delta\boldsymbol{q}^T\boldsymbol{K}_q\,\Delta\boldsymbol{q}. \tag{4.3}$$

Here, $\boldsymbol{q}^{\mathrm{e}}$ denotes the equilibrium joint configuration.

For the workspace, several methods exist to describe the potential, e.g., Fasse and Broenink (1997), Caccavale et al. (1999), Stramigioli (2001), and Natale (2003). The main difference between the methods lies in the description of the rotational part of the potential. As shown in sec. 3.2, it is possible to describe a rotation with unit-length quaternions. The rotation matrix ${}^{\mathrm{e}}\boldsymbol{R}_{\mathrm{b}} \in SO(3)$ between the body-fixed coordinate frame (subscript b) and the equilibrium pose (superscript e) is converted to unit-length quaternion with scalar rotation angle η and unit-axis $\boldsymbol{\epsilon}$ (eq. (3.6)). Therefore, the total potential $\mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})$, expressed in the body-fixed coordinate frame, is chosen to be the sum of translational potential $\mathscr{U}_p$ and rotational potential $\mathscr{U}_\epsilon$:

$$\begin{aligned}\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon}) = &\underbrace{\frac{1}{2}\Delta\boldsymbol{p}^T\boldsymbol{K}_{\boldsymbol{p}}\,\Delta\boldsymbol{p}}_{\mathscr{U}_p} + \\ &+\underbrace{2\,\boldsymbol{\epsilon}^T\,\boldsymbol{K}_{\boldsymbol{\epsilon}}\,\boldsymbol{\epsilon}}_{\mathscr{U}_\epsilon}.\end{aligned} \tag{4.4}$$

To avoid oscillations, an appropriate damping function has to be found. If the controller damps the joint velocity $\dot{\boldsymbol{q}}$, the Rayleigh function $\mathscr{R} : \mathbb{R}^n \to \mathbb{R}$ can be represented by:

$$\mathscr{R}(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{B}_q\dot{\boldsymbol{q}}. \tag{4.5}$$

Here, the resistive element $\boldsymbol{B}_q \in \mathbb{R}^{n\times n}$ can be obtained by using the damping design method of sec. 3.4.3. First, a Cartesian damping design can be applied (Albu-Schaffer et al. (2003), eq. (3.34)). Second, the resulting Cartesian damping matrix can be mapped into the joint space and joint damping values can be assigned that act on the nullspace of $\boldsymbol{J}(\boldsymbol{q})$ (eqs. (3.35) - (3.40)).

The control wrench $\boldsymbol{F}^{\text{imp}} \in se^\star(3)$ that is supposed to minimize the potential energy function $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ can be derived by computing the differential of $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ (appendix D). The control torque $\boldsymbol{\tau}^{\text{imp}} \in \mathbb{R}^n$ is calculated with the wrench $\boldsymbol{F}^{\text{imp}}$, the partial derivative of the gravitational potential energy $\mathscr{U}_{\text{g}}(\boldsymbol{q})$ (eq. (3.1)), and the partial derivative of $\mathscr{R}(\dot{\boldsymbol{q}})$(eq. (4.5)):

$$\boldsymbol{\tau}^{\text{imp}} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{F}^{\text{imp}} - \frac{\partial\mathscr{R}(\dot{\boldsymbol{q}})}{\partial\dot{\boldsymbol{q}}} + \frac{\partial\mathscr{U}_{\text{g}}(\boldsymbol{q})}{\partial\boldsymbol{q}}. \tag{4.6}$$

Here, the centrifugal/Coriolis effects are neglected.

4.3.2 Energy transfer during interaction

The interaction between the robot and its environment can be represented by an equivalent network (Hogan, 2014; Stramigioli, 2001)(fig. 4.10). For a given joint configuration and desired impedance behavior, the robot controller calculates a desired control wrench. The contact between robot and human takes place at the interaction port $(\boldsymbol{\xi}, \boldsymbol{F}^{\text{imp}})$. In a collision, the energy that is flowing bi-directionally between the robot and the human can be modeled at this port. If the robot behaves like a mechanical impedance[3], its output (seen from the robot side) is a force (effort) and the input is a displacement (time-integral of flow). The connected subsystem (in this case the human) is exposed to this force $\boldsymbol{F}^{\text{imp}}$ and influences the robot displacement.

In a contact scenario, the amount of transferred energy is determined by the energy of the robotic system, i.e., kinetic co-energy $\mathscr{T} : \mathbb{R}^n \to \mathbb{R}$ and potential

[3] Since humans are soft, the robot behaves like an admittance (as seen from the human side).

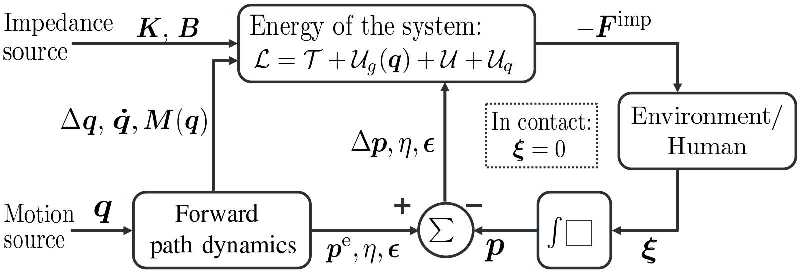

Impedance source
$\boldsymbol{K}, \boldsymbol{B}$
Energy of the system:
$\mathcal{L} = \mathcal{T} + \mathcal{U}_g(\boldsymbol{q}) + \mathcal{U} + \mathcal{U}_q$
$-\boldsymbol{F}^{\text{imp}}$
$\Delta\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{M}(\boldsymbol{q})$
$\Delta\boldsymbol{p}, \eta, \boldsymbol{\epsilon}$
In contact: $\boldsymbol{\xi} = 0$
Environment/ Human
Motion source
$\boldsymbol{q}$
Forward path dynamics
$\boldsymbol{p}^{\text{e}}, \eta, \boldsymbol{\epsilon}$
+
−
Σ
$\boldsymbol{p}$
$\int\square$
$\boldsymbol{\xi}$

Figure 4.10: Norton equivalent network for contact dynamics between robot and human (inspired by Hogan (2014)). The robot is modeled as an impedance (left). A collision can be described at the interaction port $(-\boldsymbol{F}^{\text{imp}}, \boldsymbol{\xi})$: The robot exerts a force, while the human is influencing the robot position (right). During transient contact, the energy flow is mainly determined by the energy in the robotic system, i.e., potential and kinetic co-energy. During quasi-static contact, $\boldsymbol{\xi} = 0$ while $\boldsymbol{F}^{\text{imp}}$ increases.

energy $\mathscr{U}(\Delta\boldsymbol{q})+\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})+\mathscr{U}_g(\boldsymbol{q})$. $\mathscr{T}$ is a function of the robot mass $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ and the joint velocity $\dot{\boldsymbol{q}}$. Hence, for a kinematically redundant robot, $\mathscr{T}$ can vary with the different nullspace motions. $\mathscr{U}_g(\boldsymbol{q})$ depends on the robot configuration. For pHRI, $\mathscr{U}_g(\boldsymbol{q})$ must always be compensated by the controller to eliminate the risk due to a collapsing robot (fig. 3.2). The total energy $\mathscr{L} \in \mathbb{R}$ of the robotic system is:

$$\begin{aligned} \mathscr{L} = \mathscr{T}(\dot{\boldsymbol{q}}, \boldsymbol{M}(\boldsymbol{q})) + \mathscr{U}(\Delta\boldsymbol{q}) + \\ \mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon}) \ + \ \mathscr{U}_g(\boldsymbol{q}). \end{aligned} \tag{4.7}$$

Hereby, $\boldsymbol{q}$, $\boldsymbol{p}$, and $\boldsymbol{\epsilon}$ are usually functions of the control time $t_{\text{c}} \in \mathbb{R}$.

For an impedance controlled robot, the energy flow from the information domain (impedance source $\boldsymbol{K}, \boldsymbol{B}$) to the energy domain (physical interaction $(\boldsymbol{\xi}, \boldsymbol{F}^{\text{imp}})$) follows a strict causality. First, the robot is exposed to physical and virtually controlled potential energy $\mathscr{U}(\Delta\boldsymbol{q}) + \mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon}) + \mathscr{U}_g(\boldsymbol{q})$. Once the robot starts moving, the potential energy is transferred to kinetic co-energy. In case of a collision, $\mathscr{T}$ is exchanged with the human. Since the skin is deformed, the contact can be modeled as an energy transfer from $\mathscr{T}$ to $\mathscr{U}_{\text{skin}} \in \mathbb{R}$ (International Organization for Standardization, 2016; Haddadin et al., 2011). In a quasi-static contact, $\mathscr{T} \approx 0$ and the human is only affected by the potential energy, which generates a clamping force at the interaction port. The controlled motion and constant robot position $\boldsymbol{p}$ leads to an increase of $\mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ and therefore to an increase of the clamping wrench $\boldsymbol{F}^{\text{imp}}$.

In the following, an energy budget will be assigned to the robot system in order to ensure an intrinsically safe robot motion and safe contact behavior. Therefore, the wrench $\boldsymbol{F}^{\text{imp}}$ is automatically limited in case of a clamping scenario.

4.3.3 Energy budget for safe pHRI

If $n > 6$, $\mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})$ does not affect the nullspace in the $\boldsymbol{J(q)}$. This nullspace exists for a kinematically redundant robot and–even if damped–can lead to an increase of $\mathscr{T}$ due to nullspace motion. Consequently, if the robot energy should be controlled, not only potential energy but also $\mathscr{T}$ has to be taken into account. For simplification, $\mathscr{U}_g(\boldsymbol{q})$ will be excluded from the following equations since it always has to be compensated by the controller, e.g., via energy shaping techniques (Duindam et al., 2009). The controlled energy $\mathscr{L}_\mathrm{c} \in \mathbb{R}$, expressed in body-fixed coordinate frame, can be calculated with

$$\begin{aligned}\mathscr{L}_\mathrm{c} = &\underbrace{\frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M(q)}\dot{\boldsymbol{q}}}_{\mathscr{T}} + \underbrace{\frac{1}{2}\Delta\boldsymbol{q}^T \boldsymbol{K_q}\ \Delta\boldsymbol{q}}_{\mathscr{U}(\Delta\boldsymbol{q})} + \\ &+ \underbrace{\frac{1}{2}\Delta\boldsymbol{p}^T \boldsymbol{K_p}\ \Delta\boldsymbol{p}}_{\mathscr{U}_p} + \underbrace{2\ \boldsymbol{\epsilon}^T\ \boldsymbol{K_\epsilon}\ \boldsymbol{\epsilon}}_{\mathscr{U}_\epsilon}.\end{aligned} \tag{4.8}$$

The presented controller will set an energy budget $\mathscr{L}_\mathrm{max} \in \mathbb{R}$ for the robot, satisfying

$$\mathscr{T} + \mathscr{U}(\Delta\boldsymbol{q}) + \mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon}) \leq \mathscr{L}_\mathrm{max}. \tag{4.9}$$

If eq. (4.9) is violated, a new elastic potential $\mathscr{U}^\star(\Delta\boldsymbol{p}, \boldsymbol{\epsilon}) = \kappa\ \mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})$ and $\mathscr{U}^\star(\Delta\boldsymbol{q}) = \kappa\ \mathscr{U}(\Delta\boldsymbol{q})$ can be produced with the scaling function $\kappa : \mathbb{R} \to \mathbb{R}$:

$$\kappa = \begin{cases} 1 & ,\text{if } \mathscr{L}_\mathrm{c} \leq \mathscr{L}_\mathrm{max} \\ \dfrac{\mathscr{L}_\mathrm{max} - \mathscr{T}}{\mathscr{U}(\Delta\boldsymbol{q}) + \mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})} & ,\text{if } \mathscr{L}_\mathrm{c} > \mathscr{L}_\mathrm{max} \\ 0 & ,\text{if } \mathscr{T} > \mathscr{L}_\mathrm{max}. \end{cases} \tag{4.10}$$

This approach will automatically bound the total energy of the robot. The condition for $\mathscr{T} > \mathscr{L}_\mathrm{max}$ is necessary for pHRI: If the human were to push the robot while the scaling function was active, this could result in a high negative κ value, which would lead to unstable robot behavior. A simpler form of eq. (4.10) was used in Tadele et al. (2014) and Raiola et al. (2018). This work provides two necessary extensions.

First, the control inputs $\boldsymbol{p}^\mathrm{e}$ and ${}^\mathrm{b}\boldsymbol{R}_\mathrm{e}$ will be changed to be a function of what will be called "effective time," $t_\mathrm{c} \to t_\mathrm{eff}$. If no collision happens, $t_\mathrm{c} = t_\mathrm{eff}$ and t_eff will be updated by the sample time of the controller $t_\mathrm{s} \in \mathbb{R}$ at the end of each control cycle. Otherwise, if a contact occurs, the controller does not update t_eff and therefore remains in the pose at the contact incident. The time t_eff can be calculated by:

$$\mathbb{R} \ni t_\mathrm{eff} = \begin{cases} t_\mathrm{eff} = t_\mathrm{eff} + t_\mathrm{s} & ,\text{if } \mathscr{L}_\mathrm{c} \leq \mathscr{L}_\mathrm{max} \\ t_\mathrm{eff} = t_\mathrm{eff} & ,\text{otherwise.} \end{cases} \tag{4.11}$$

This is a necessary extension, since otherwise the time-dependent control variables $\boldsymbol{p}^\mathrm{e}$ and ${}^\mathrm{b}\boldsymbol{R}_\mathrm{e}$ would be updated during contact. After the contact, the potential energy $\mathscr{U}(\Delta\boldsymbol{q}) + \mathscr{U}(\Delta\boldsymbol{p},\boldsymbol{\epsilon})$ would be very high due to the large displacement between end-effector pose and equilibrium pose. The energy budget would continue to be violated and the scaling function κ would continue to act, even though the robot was not in a contact scenario anymore. This would lead to an undesired and unpredictable robot motion.

Second, an additional joint potential $\mathscr{U}^\bullet(\Delta\boldsymbol{q}) \in \mathbb{R}$ will be introduced:

$$\mathscr{U}^\bullet(\Delta\boldsymbol{q}) = \begin{cases} 0 & \text{, if } \mathscr{T} \leq \mathscr{L}_{\max} \\ \frac{1}{2}(\boldsymbol{q}-\boldsymbol{q}^\bullet)^T \; \boldsymbol{K}_q{}^\bullet \; (\boldsymbol{q}-\boldsymbol{q}^\bullet), & \text{, if } \mathscr{T} > \mathscr{L}_{\max}, \end{cases} \tag{4.12}$$

with $\boldsymbol{q}^\bullet \in \mathbb{R}^n$ being the joint configuration of the collision incident ($\mathscr{T} > \mathscr{L}_{\max}$), which is stored until $\mathscr{T} \leq \mathscr{L}_{\max}$. This is the case if the operator pushes the robot, e.g., when releasing from a clamping scenario. The controller will be extended with a second scaling function $\rho : \mathbb{R} \to \mathbb{R}$, producing a new joint potential $(\rho\ \mathscr{U}^\bullet(\Delta\boldsymbol{q}))$:

$$\rho = \begin{cases} 1 & \text{, if } \mathscr{T} \leq \mathscr{L}_{\max} \\ \Omega \; \dfrac{\mathscr{T}}{\mathscr{L}_{\max} + \mathscr{U}^\bullet(\Delta\boldsymbol{q})} & \text{, if } \mathscr{T} > \mathscr{L}_{\max}, \end{cases} \tag{4.13}$$

where $\Omega \geq 1$ is a scalar value that determines the desired robot behavior. With eqs. (4.12) and (4.13), the robot increases the joint stiffness if $\mathscr{T}$ would violate the energy budget $\mathscr{L}_{\max}$. This is a necessary feature since otherwise the joints could accelerate without control if the robot was pushed. The effect of this feature can be seen in the multimedia extension (summary part V) of Lachner et al. (2021).

Finally, the control torque $\boldsymbol{\tau}^{\text{imp}}$ can be calculated with $\boldsymbol{F}^{\text{imp}}$ based on appendix D, with the partial derivatives of $\mathscr{U}(\Delta\boldsymbol{q})$ (eq. (4.3)), $\mathscr{R}(\dot{\boldsymbol{q}})$ (eq. (4.5)), $\mathscr{U}^\bullet(\Delta\boldsymbol{q})$ (eq. (4.12)), and by application of the energy scaling method of eqs. (4.10) and (4.13):

$$\begin{aligned} \boldsymbol{\tau}^{\text{imp}} &= \boldsymbol{J}(\boldsymbol{q})^T \left(\kappa \; \boldsymbol{F}^{\text{imp}}\right) + (\kappa \; \frac{\partial \mathscr{U}(\Delta\boldsymbol{q})}{\partial \Delta\boldsymbol{q}}) - (\sqrt{\kappa} \; \frac{\partial \mathscr{R}(\dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}) + \rho \; \frac{\partial \mathscr{U}^\bullet(\Delta\boldsymbol{q})}{\partial(\boldsymbol{q}-\boldsymbol{q}^\bullet)} = \\ &= \boldsymbol{J}(\boldsymbol{q})^T \begin{pmatrix} -\,\kappa\;\boldsymbol{K}_p\;\Delta\boldsymbol{p} \\ \Delta\tilde{\boldsymbol{p}}\;\kappa\;\boldsymbol{K}_p\;\Delta\boldsymbol{p} + 2\;\boldsymbol{E}(\eta,\boldsymbol{\epsilon})^T\;\kappa\;\boldsymbol{K}_\epsilon\;\boldsymbol{\epsilon} \end{pmatrix} + \kappa\;\boldsymbol{K}_q\;\Delta\boldsymbol{q} \\ &\quad - \sqrt{\kappa}\;\boldsymbol{B}_q\;\dot{\boldsymbol{q}} + \rho\;\boldsymbol{K}_q{}^\bullet(\boldsymbol{q}-\boldsymbol{q}^\bullet). \end{aligned} \tag{4.14}$$

Here, $\boldsymbol{E}(\eta,\boldsymbol{\epsilon})$ can be calculated based on eq. (D.11). Note that for $\mathscr{T} < \mathscr{L}_{\max}$: $\boldsymbol{q}^\bullet = \boldsymbol{q}$ and hence $\mathscr{U}^\bullet(\Delta\boldsymbol{q}) = 0$.

A great benefit of this general approach is that the scaling functions can be applied no matter how the elastic potential is described. Hence, the presented controller is a feasible extension for every impedance controller. Moreover, in the presented version, the scaling parameter κ also affects the damping design in eq. (3.34) and the desired damping behavior is also ensured if $\kappa < 1$. With the control approach, not only the potential energy can be limited during quasi-static contact, but also the kinetic co-energy if the robot is pushed.

4.3.4 Derivation of a safe energy budget

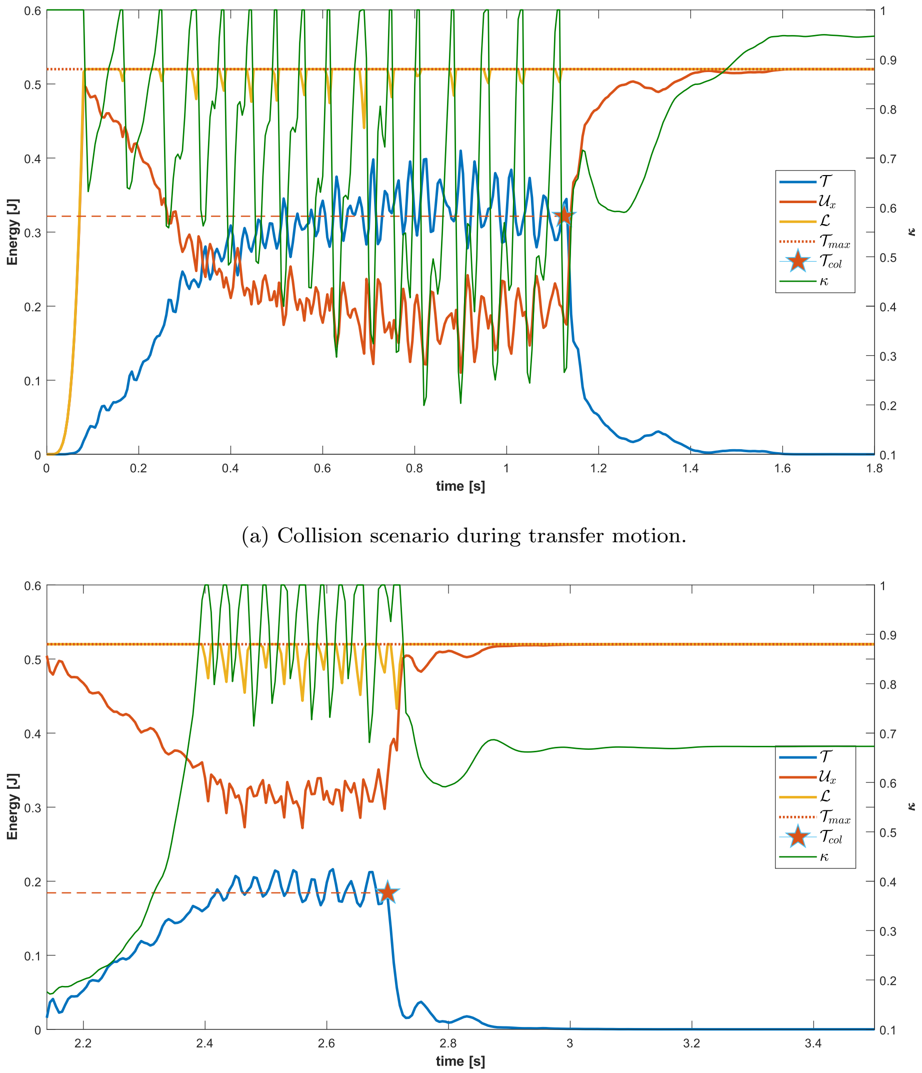
ISO/TS 15066 provides energy thresholds that can be calculated via

$$\mathscr{L}_{\max} = \frac{F_{\max}^2}{2k}. \tag{4.15}$$

With the minimal force values and stiffness parameters of tab. 4.1, the energy thresholds can be calculated for the example application of fig. 4.2 (tab. 4.2). These values determine the limit to the bio-mechanical energy that can be exchanged during the transient ($\mathscr{L}_{\mathrm{TF}}$) and quasi-static ($\mathscr{L}_{\mathrm{QSF}}$) contact phases. First, the transient contact phase will be analyzed and $\mathscr{L}_{\max} = 0.52$ J will be set for the controller. All other control parameters can be seen in appendix G.2. In the following, the same collision measurements as in sec. 4.2.1 were performed. As can be seen in fig. 4.11, the robot automatically limits $\mathscr{L}_{\mathrm{c}} \leq 0.52$ J. Once $\mathscr{L}_{\mathrm{c}}$ reaches $\mathscr{L}_{\max}$, the scaling function κ (eq. (4.10)) acts on the impedance potential $\mathscr{U}(\Delta\boldsymbol{p}, \boldsymbol{\epsilon})$ and consequently reduces the kinetic co-energy $\mathscr{T}$ of the robot.

Table 4.2: Bio-mechanical energy thresholds based on ISO/TS 15066 for affected body parts in the example application. The bold numbers indicate the lowest values that have to be taken into account.

	$\mathscr{L}_{\mathrm{TF}}$ [J]	$\mathscr{L}_{\mathrm{QSF}}$ [J]
Up. arm	1.50	0.38
Lo. arm	1.28	0.32
Hand	**0.52**	**0.13**

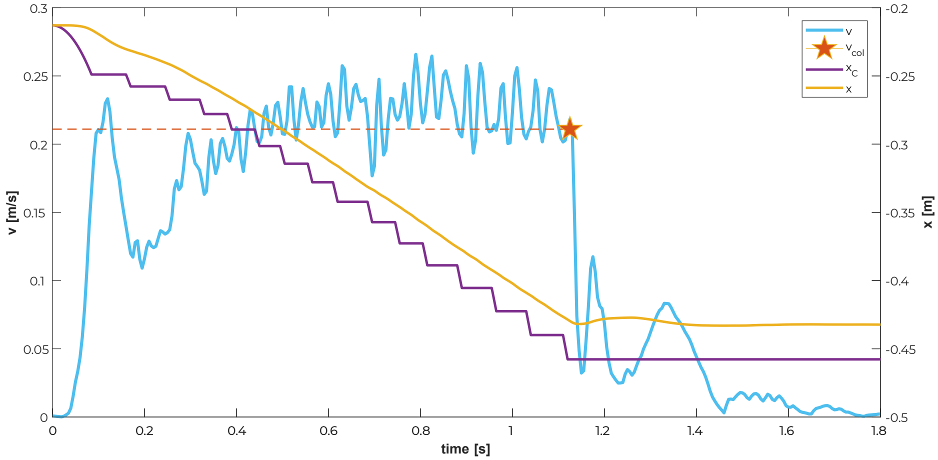


(a) Collision scenario during transfer motion.

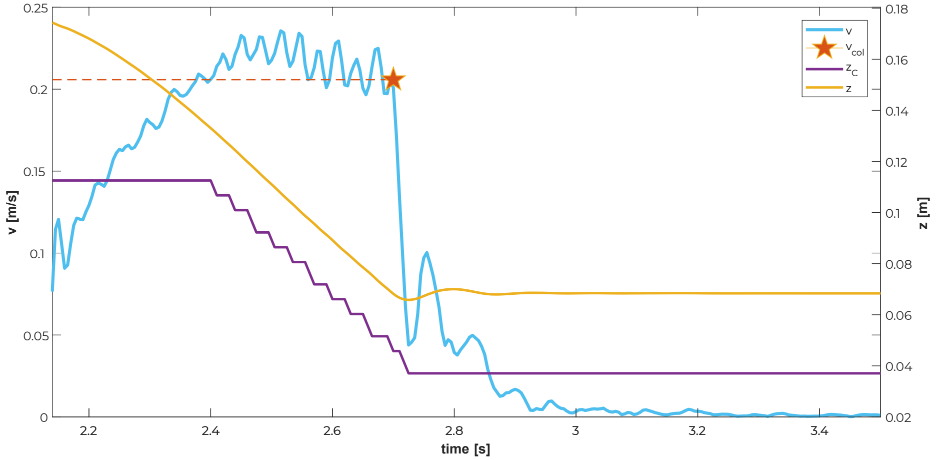
(b) Collision scenario during approach motion.

Figure 4.11: Energy distribution with active scaling function during transfer motion (a) and approach motion (b). Once the threshold $\mathscr{L}_{\max}$ is reached, the scaling function κ acts on the impedance potential $\mathscr{U}_x = \mathscr{U}(\Delta \boldsymbol{p}, \boldsymbol{\epsilon})$ and consequently reduces $\mathscr{T}$. After the first collision instant (red star), $\mathscr{T}$ reduces quickly and $\mathscr{U}_x$ approaches $\mathscr{L}_{\max}$.

Moreover, as long as $\mathscr{L}_{\mathrm{c}} > \mathscr{L}_{\max}$, the controlled position (x- and z-coordinate) is kept constant (fig. 4.12). Note that if solely the controlled robot pose is kept constant, the energy constraint $\mathscr{L}_{\max}$ may be violated. This can be seen in the example application during quasi-static contact with $\kappa < 1$ (fig. 4.11) and static x/z-coordinate (fig. 4.12).



(a) Collision scenario during transfer motion.



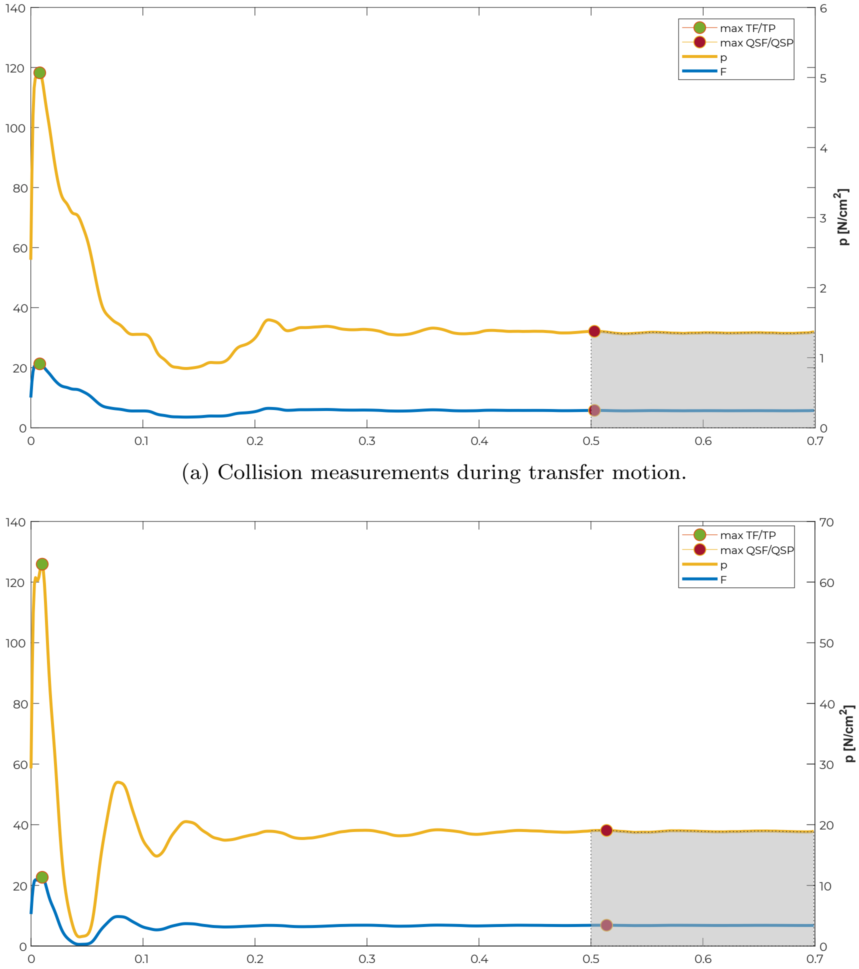
(b) Collision scenario during approach motion.

Figure 4.12: Velocity and position with active scaling during transfer motion (a) and approach motion (b). The robot velocity quickly reduces after the first collision instant (v_{col}). During the transfer motion, the robot moves along the x-coordinate of the base frame. During the approach motion, the robot moves along the z-coordinate of the 0-coordinate frame. The controlled robot position is kept constant if $\mathscr{L}_{\mathrm{c}} > \mathscr{L}_{\mathrm{max}}$ (purple line).

During the transient contact phase, $\mathscr{T}$ is exchanged with the human and is hence the crucial quantity to observe. To comply with ISO/TS 15066, it is necessary that $\mathscr{T} \leq 0.52$ J during this phase. For the novel control approach, however, the robot will never reach this threshold since there will always be a potential energy content in $\mathscr{L}_\mathrm{c}$ (cf., $\mathscr{T}$ in fig. 4.11).

During the quasi-static contact phase, the potential energy imposes a clamping force. The calculated energy threshold based on ISO/TS 15066 for this phase is 0.13 J. It could be shown that the novel control approach eliminates the clamping risk. This can be seen by observing the QSF values during the transfer motion and QSP values during the approach motion in the example application. While for the classical approach of sec. 4.2.1, the QSF values after the robot brakes were activated remained approximately 65 N (fig. 4.6b), these values were limited to ≤ 6.86 N with the novel approach (fig. 4.13a). The same holds for the QSP values after a collision during the approach motion: While for the classical approach a constant pressure of approximately 180 N/cm^2 remained (fig. 4.6b), this value was limited to ≤ 19.06 N/cm^2 with the novel approach (fig. 4.13b). For the example application, it can therefore be argued that clamping dangers don't have to be taken into account and the double force values can be used to calculate $\mathscr{L}_\mathrm{QSF}$. This is the same value as for the transient contact phase: $\mathscr{L}_\mathrm{max} = \mathscr{L}_\mathrm{TF} = \mathscr{L}_\mathrm{QSF} = 0.52$ J. Hence, the presented controller is especially advantageous for approach motions where the tool tip usually has a small surface area.

The classical approach showed large differences between measurements with rounded and sharp objects. During the transient contact phase with gripper cover and gripper jaw, the maximum transient force values were 268.02 N and 100.42 N, respectively. During the quasi-static contact phase with the same colliding parts, the maximum QSP values were 22.56 N/cm^2 and 180.73 N/cm^2 (fig. 4.6b). The novel control approach yields comparable force peaks during the transient phase (21.28/22.66 N) and similar maximum force values during the quasi-static phase (5.78/6.86 N) (cf. fig. 4.13a and 4.13b). Even though the surface area of the gripper jaws was small, the pressure values were reduced to ≤ 19.06 N/cm^2 (fig. 4.13b). While every segment of a robot application has to be manually adapted using the classical approach, the novel controller constrains the robot to a consistent behavior during the whole application. As can be seen in fig. 4.13, the controller is fast enough to minimize the quasi-static contact values within 0.5 seconds of the transient contact phase. Moreover, instead of adapting multiple control parameters in the classical approach (e.g., $\boldsymbol{\tau}_\mathrm{max}$, v_max, $\boldsymbol{K}_q$, $\boldsymbol{K}_p$, and $\boldsymbol{K}_\epsilon$), the approach presented here requires only one parameter $\mathscr{L}_\mathrm{max}$ and auto-tunes the initial control parameters to guarantee a safe robot behavior.



(a) Collision measurements during transfer motion.

(b) Collision measurements during approach motion.

Figure 4.13: Force and pressure distribution with active scaling function during transfer motion (a) and approach motion (b). The maximal transient force values (21.28/22.66 N) and quasi-static force values (5.78/6.86 N) are in a comparable range for the transfer and approach motions. All values are far lower than the permitted bio-mechanical thresholds. In practice, the energy budget could be iteratively adapted and verified based on collision measurements.

4.4 Conclusion

The first aim of this chapter was to make the reader aware of the influence of coordinates on the implementation of safety measures in pHRI applications. The current certification process was reviewed with an example application on a real robot. Coordinate frames had to be placed on the robot structure to monitor

Cartesian velocities. With collision measurements, the velocity was iteratively adapted until no bio-mechanical thresholds were violated. For a poor choice of coordinate frame positions, unmonitored high velocities present risks if unintended collisions occur. This has especially to be considered for kinematically redundant robots, for which a motion in the nullspace of $\boldsymbol{J}(\boldsymbol{q})$ can occur. If joint torque sensors are used to detect collisions, the reliability of the detection depends on the current robot configuration.

The second aim of this chapter was to present a novel approach that removed these limitations by assigning an energy budget for impedance controlled robots. This coordinate invariant control approach facilitates the certification process of pHRI applications. The key features of this control approach are:

1. The energy budget is a coordinate invariant quantity that acts on the whole robot. No coordinate frames have to be placed on the robot structure. For kinematically redundant robots, fast nullspace motions are automatically reduced.

2. The detection of collisions is independent of the current robot configuration. No external sensors are needed.

3. The controller can be used for all contact scenarios. The transferred energy during transient contact is limited to a maximal threshold. This energy scaling method eliminates high static forces in clamping scenarios.

4. During contact, the robot is compliant. If the robot is pushed, its energy threshold is still guaranteed since an additional joint potential prevents too high kinetic co-energy. After contact, the robot automatically continues to move on its pre-planned trajectory.

5. This new approach facilitates the current certification process: Instead of adapting multiple control parameters, the presented approach has one control input $\mathscr{L}_{\max}$ and auto-tunes the other control parameters. The experiments in this paper used a conservative value for $\mathscr{L}_{\max}$. In practice, $\mathscr{L}_{\max}$ can be optimized for each part of the robot application based on collision measurements. In the presented application, this is especially advantageous to speed up transfer motions.

4.4.1 Limitations

A practical limitation of the presented approach is its need for accurate dynamic model data for the robot. Without access to $\boldsymbol{M}(\boldsymbol{q})$ of the robot, $\mathscr{T}$ cannot be calculated accurately. However, multiple methods exist to identify $\boldsymbol{M}(\boldsymbol{q})$ (cf. the overview provided in Siciliano and Khatib (2008)). Even with that limitation, the control approach can still be used for quasi-static contact, since $\mathscr{T} \approx 0$ and $\mathscr{U}(\Delta \boldsymbol{p}, \boldsymbol{\epsilon})/\mathscr{U}(\Delta \boldsymbol{q})$ can be calculated without dynamic model data.

The energy threshold $\mathscr{L}_{\max}$ defined in ISO/TS 15066 can also be calculated based on the minimal pressure values and stiffness parameters of tab. 4.1:

$$\mathscr{L}_{\max} = \frac{A^2 P_{\max}^2}{2k}. \tag{4.16}$$

In comparison to eq. (4.15), where $\mathscr{L}_{\max}$ included forces, $\mathscr{L}_{\max}$ in eq. (4.16) depends on the contact surface area $A \in \mathbb{R}$. Selecting the gripper jaw surface and the minimal pressure thresholds of the lower arm would yield $\mathscr{L}_{\mathrm{TF}} = 0.21$ J for the transient contact phase and $\mathscr{L}_{\mathrm{QSF}} = 0.05$ J for the quasi-static contact phase. In contrast to these calculated thresholds, it was shown in sec. 4.3.4 that setting $\mathscr{L}_{\max} = 0.52$ J yields far lower QSF and QSP values than allowed by ISO/TS 15066. For a very sharp tool, however, this energy threshold might be too high. Hence, it can be concluded that one cannot solely rely on the calculations based on ISO/TS 15066 and that it is still necessary to certify the selected energy thresholds for each part of the robot application via collision measurements.

Many pHRI applications involve contact, e.g., during assembly processes. These processes might involve potential energy greater than $\mathscr{L}_{\max}$. In case of a clamping scenario before the robot is in contact, it has to be ensured that the bio-mechanical thresholds are not violated. Hence, strategies have to be applied to identify the contact phase (Bicchi et al., 1993) and the energy budget has to be adapted accordingly.

Lightweight robots have low inertia and hence low kinetic co-energy during movement, which is a beneficial factor for the transient contact phase (De Santis et al., 2008). Therefore, the presented work is a promising control approach for these kind of robots. Classical industrial robots, however, move high masses and therefore have limited application to pHRI. For these robots, external safety measures are needed that monitor the distance between the robot and the human or separate their respective workspaces.

4.4.2 Towards safe pHRI in changing work environments

In case the safe distance is violated or the human enters the robot workspace, classical industrial robots have to stop as quickly as possible. Since clamping risks are excluded, this is a safe reaction. For pHRI applications, however, extensive care has to be taken, to minimize the contact force and pressure during clamping scenarios with the stopped robot. The programming of such applications is much more sophisticated since the correct placement of the coordinate frames is crucial for safety. Moreover, the importance of the risk assessment increases, e.g., to identify all affected body parts. For unstructured environments, the author claims that it is impossible to predict all risks in advance. Due to the drawbacks of coordinate-based safety methods shown in this chapter, it should be emphasize to focus on intrinsic safety mechanisms for pHRI. The author therefore encourages one to review and revise the definition of a safe reaction in current standards and regulations, e.g., in ISO/TS 15066, ISO 10218-1, and ISO 10218-2.

CHAPTER 5

General Conclusion

This thesis provides a geometric approach for the analysis and control of kinematically redundant robots. By minimizing the coordinate dependency, the proposed methods improve the robot behavior during physical Human-Robot Interaction. The key contributions of this thesis include:

- An analysis of the coordinate dependency in robotic dexterity and a proposal of a dexterity measure that can be applied to all kinematic structures, including robots with mixed joint types.
- An extension for a Cartesian impedance controller, which makes use of the proposed dexterity measure and enables multi-task control, while preserving passivity.
- A novel control approach that facilitates the programming and certification process of pHRI applications and ensures safety through a coordinate invariant approach.

This work postulates that collaborative robots have to comply with the surrounding physics: Firstly, to make general statements about the robot's abilities, the analysis must not depend on specific coordinate choices, made by the robot programmer; secondly, physical attributes have to be assigned to the control parameters; thirdly, the causality of the robot behavior during physical interaction has to be respected, which is mainly dictated by energy exchange.

The provided control frameworks make a step towards easy-to-use collaborative robots since the control parameters are auto-tuned during runtime to ensure performance, stability, and safety. This not only enables programming by inexperienced robotic engineers, but also significantly speeds up the commissioning of such robot cells.

During physical interaction, the robot behavior is dictated by its own dynamics and the dynamics of the surrounding environment. By definition, dynamic robot models are a simplification of reality. It is even harder to derive a competent model of the environment. Therefore, simple impedance control is a beneficial control approach since it renders the dynamics without involving the inertia matrix. This work reduced the Cartesian error of a simple impedance controller at the cost of the need for a competent dynamic robot model.

The parallel computation of impedance and motion in human motor planning was the main inspiration for the impedance shaping approach (chapter 3). However, humans do not re-calculate their desired impedance behavior cyclically, but utilize sets of impedance parameters which were learned while growing up. Future work should combine the methodology of this thesis with new advances in learning and optimization.

Future work can merge the control approaches of chapters 3 and 4. Even though chapter 4 did not specifically address the passivity property of the controller, chapter 3 showed a solution for how to handle the power flow between the controller and the robot. Therefore, all time-varying components of the energy budget method can be identified and the passivation method of sec. 3.4.6 can be applied.

The proposed methodologies in this thesis can find various applications in industry to facilitate the planning and programming of collaborative robot cells. Some of the applications might include:

- At present, customers have to consult robotic experts to find an appropriate robot for their automation application. The dexterity measure derived in chapter 2 can be applied in an online tool and empowers the customer to find the right robot for a given application.

- Even though an impedance-controlled robot can go in and out of singularities without control issues, the end-effector accuracy might suffer and the kinematic structure might be subject to high stress. The control approach of chapter 3 does not only reduce this issue, but also enables multi-task control which is not common yet in industrial robotics.

- By assigning an energy budget to the robot (chapter 4), especially the safety for unstructured environments can be enhanced. Moreover, such a controller is beneficial during the programming phase of a robot cell since the robot can be stopped at each time instant. This can reduce collision hazards with the environment due to programming errors.

Appendices

APPENDIX A

Dexterity Analysis

A.1 $det(J(q)J(q)^T)$ for different body coordinate frames

The Hybrid Jacobian Matrix $\boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ maps joint velocities to spatial velocities which are expressed in the end-effector coordinate frame “a” with respect to the inertial coordinate frame “0.” If another body-fixed coordinate frame “c” should be chosen, the Jacobian matrix $\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ can be calculated by:

$$\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q}) = \boldsymbol{Ad}^{-1}_{{}^{\mathrm{a}}H_{\mathrm{c}}} \boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q}), \tag{A.1}$$

where $\boldsymbol{Ad}^{-1}_{{}^{\mathrm{a}}H_{\mathrm{c}}} \in \mathbb{R}^{6\times 6}$ is the inverse of the “Adjoint matrix” (Murray et al., 1994):

$$\boldsymbol{Ad}_{{}^{\mathrm{a}}H_{\mathrm{c}}} = \begin{pmatrix} {}^{\mathrm{a}}\boldsymbol{R}_{\mathrm{c}} & {}^{\mathrm{a}}\tilde{\boldsymbol{p}}_{\mathrm{c}}\ {}^{\mathrm{a}}\boldsymbol{R}_{\mathrm{c}} \\ 0 & {}^{\mathrm{a}}\boldsymbol{R}_{\mathrm{c}} \end{pmatrix}. \tag{A.2}$$

Here, ${}^{\mathrm{a}}\boldsymbol{R}_{\mathrm{c}} \in SO(3)$ and ${}^{\mathrm{a}}\tilde{\boldsymbol{p}}_{\mathrm{c}} \in \mathbb{R}^3$ are the rotation matrix and the position (in skew-symmetric matrix form) between “c” and “a.” The product $\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})^T$ can be expressed as:

$$\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})^T = \boldsymbol{Ad}^{-1}_{{}^{\mathrm{a}}H_{\mathrm{c}}} \boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q})\boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q})^T \boldsymbol{Ad}^{-T}_{{}^{\mathrm{a}}H_{\mathrm{c}}}. \tag{A.3}$$

Considering that $det(\boldsymbol{R}) = 1$, $det(\tilde{\boldsymbol{p}}) = 0$, and therefore $det(\boldsymbol{Ad}^{-1}_{H}) = det(\boldsymbol{Ad}^{-T}_{H}) = 1$, it can be concluded:

$$det(\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J}^{\mathrm{c}}(\boldsymbol{q})^T) = det(\boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q})\boldsymbol{J}^{\mathrm{a}}(\boldsymbol{q})^T). \tag{A.4}$$

This shows that changing the body-fixed coordinate frame doesn’t change the result of $det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)$.

A.2 $J(q)J(q)^T$ for different sets of joint coordinates

Figure 2.2 and fig. 2.3 show a 3-DOF planar robot with parameters:

- $l_1 = l_2 = l_3 = 1\ m$ (Link lengths)
- $p_{I_1} = p_{I_2} = p_{I_3} = 0.5\ m$ (Center of gravity)
- $m_1 = m_2 = m_3 = 1\ kg$ (Link masses)
- $I_1 = I_2 = I_3 = 1\ kgm^2$ (Inertia of links),

different sets of joint coordinates:

- Joint coordinate set 1: $\theta_1 = \frac{\pi}{9}\ rad$, $\theta_2 = \frac{\pi}{4}\ rad$, $\theta_3 = \frac{\pi}{3}\ rad$
- Joint coordinate set 2: $p_1 = 0.34\ m$, $p_2 = 0.71\ m$, $p_3 = 0.87\ m$
- Joint coordinate set 3: $\theta_1 = \frac{\pi}{9}\ rad$, $\theta_2 = \frac{\pi}{4}\ rad$, $p_3 = 0.87\ m$,

and randomly chosen weighting matrix $\boldsymbol{W}_{\text{rand}} = \begin{pmatrix} 3.00 & 0.00 & 0.00 \\ 0.00 & 0.3 & 0.00 \\ 0.00 & 0.00 & 0.03 \end{pmatrix}$. The Hybrid Jacobian Matrices for joint coordinate set $1-3$ in the given configuration are:

- $\boldsymbol{J(q)}_1 = \begin{pmatrix} -2.07\ \frac{m}{rad} & -1.73\ \frac{m}{rad} & -0.82\ \frac{m}{rad} \\ 0.79\ \frac{m}{rad} & -0.15\ \frac{m}{rad} & -0.57\ \frac{m}{rad} \end{pmatrix}$
- $\boldsymbol{J(q)}_2 = \begin{pmatrix} -2.20 & -2.44 & -1.64 \\ 0.84 & -0.21 & -1.15 \end{pmatrix}$
- $\boldsymbol{J(q)}_3 = \begin{pmatrix} -2.07\ \frac{m}{rad} & -1.73\ \frac{m}{rad} & -1.64 \\ -0.79\ \frac{m}{rad} & -0.15\ \frac{m}{rad} & -1.15 \end{pmatrix}$.

Consequently, the product $\boldsymbol{J(q)J(q)^T}$ for each joint coordinate set yields:

- $\boldsymbol{J(q)J(q)}_1^{\boldsymbol{T}} = \begin{pmatrix} 7.92\ \frac{m^2}{rad^2} & -0.90\ \frac{m^2}{rad^2} \\ -0.90\ \frac{m^2}{rad^2} & 0.97\ \frac{m^2}{rad^2} \end{pmatrix}$
- $\boldsymbol{J(q)J(q)}_2^{\boldsymbol{T}} = \begin{pmatrix} 13.48 & 0.55 \\ 0.55 & 2.07 \end{pmatrix}$
- $\boldsymbol{J(q)J(q)}_3^{\boldsymbol{T}} = \begin{pmatrix} 7.28\frac{m^2}{rad^2} + 2.69 & -1.38\frac{m^2}{rad^2} + 1.89 \\ -1.38\frac{m^2}{rad^2} + 1.89 & 0.65\frac{m^2}{rad^2} + 1.32 \end{pmatrix}$.

The respective eigenvalues and eigenvectors of $\boldsymbol{J(q)J(q)}_1^{\boldsymbol{T}}$ and $\boldsymbol{J(q)J(q)}_2^{\boldsymbol{T}}$ are different:

- Eigenvalues of $\boldsymbol{J(q)J(q)}_1^{\boldsymbol{T}}$: $\lambda_1 = 0.86, \lambda_2 = 8.04$
- (Right) Eigenvectors of $\boldsymbol{J(q)J(q)}_1^{\boldsymbol{T}}$: $u_1 = \begin{pmatrix} -0.13 \\ -0.99 \end{pmatrix}, u_2 = \begin{pmatrix} -0.99 \\ 0.13 \end{pmatrix}$

- Eigenvalues of $\boldsymbol{J(q)J(q)}_2^{\boldsymbol{T}}$: $\lambda_1 = 13.51, \lambda_2 = 2.04$
- (Right) Eigenvectors of $\boldsymbol{J(q)J(q)}_2^{\boldsymbol{T}}$: $u_1 = \begin{pmatrix} 0.05 \\ -1.00 \end{pmatrix}, u_2 = \begin{pmatrix} -1.00 \\ -0.05 \end{pmatrix}$

The eigenvalues and the determinant of $\boldsymbol{J(q)J(q)}_3^{\boldsymbol{T}}$ cannot be calculated because a mismatch of units arises.

A.3 Quadratic form Λ^{-1}

The mass matrices for joint coordinate set $1 - 3$ in the given configuration are:

- $\boldsymbol{M(q)}_1 = \begin{pmatrix} 9.11\ \frac{kgm^2}{rad} & 4.93\ \frac{kgm^2}{rad} & 1.37\ \frac{kgm^2}{rad} \\ 4.93\ \frac{kgm^2}{rad} & 4.00\ \frac{kgm^2}{rad} & 1.50\ \frac{kgm^2}{rad} \\ 1.37\ \frac{kgm^2}{rad} & 1.50\ \frac{kgm^2}{rad} & 1.25\ \frac{kgm^2}{rad} \end{pmatrix}$
- $\boldsymbol{M(q)}_2 = \begin{pmatrix} 10.32\ kg & 7.42\ kg & 2.92\ kg \\ 7.42\ kg & 8.00\ kg & 4.24\ kg \\ 2.92\ kg & 4.24\ kg & 5.00\ kg \end{pmatrix}$
- $\boldsymbol{M(q)}_3 = \begin{pmatrix} 9.11\ \frac{kgm^2}{rad} & 4.93\ \frac{kgm^2}{rad} & 2.74\ kgm \\ 4.93\ \frac{kgm^2}{rad} & 4.00\ \frac{kgm^2}{rad} & 3.00\ kgm \\ 2.74\ kgm & 3.00\ kgm & 5.00\ kg \end{pmatrix}.$

Consequently, the matrix $\boldsymbol{\Lambda}^{-1}$ for each set of joint coordinates yields $\boldsymbol{\Lambda}_1^{-1} = \boldsymbol{\Lambda}_2^{-1} = \boldsymbol{\Lambda}_3^{-1} = \boldsymbol{J(q)}_1\boldsymbol{M(q)}_1^{-1}\boldsymbol{J(q)}_1^T = \boldsymbol{J(q)}_2\boldsymbol{M(q)}_2^{-1}\boldsymbol{J(q)}_2^T = \boldsymbol{J(q)}_3\boldsymbol{M(q)}_3^{-1}\boldsymbol{J(q)}_3^T = \begin{pmatrix} 0.79\ \frac{1}{kg} & 0.18\ \frac{1}{kg} \\ 0.18\ \frac{1}{kg} & 0.53\ \frac{1}{kg} \end{pmatrix}$. It can be seen that the mobility end-point tensor is equal for all joint coordinate choices.

A.4 Bi-invariant matrix Λ_ϵ^{-1}

Figure 2.6 shows a 8-DOF robot with joint coordinate set:
$p_1 = 0\ m$, $p_2 = 0\ m$, $\theta_1 = 0\ rad$, $\theta_2 = -\frac{pi}{2}\ rad$, $\theta_3 = \frac{pi}{3}\ rad$, $\theta_4 = -\frac{pi}{2}\ rad$, $\theta_5 = \frac{pi}{3}\ rad$, $\theta_6 = 0\ rad$,
and two different body-fixed coordinate frames Ψ_1 and Ψ_2 on end-effector body. The respective homogeneous transformation matrices ${}^0\boldsymbol{H}_1 \in SE(3)$ of Ψ_1 and ${}^0\boldsymbol{H}_2 \in SE(3)$ of Ψ_2 are:

- ${}^0\boldsymbol{H}_1 = \begin{pmatrix} 0.00 & 1 & 0 & 0.17\ m \\ -0.87 & 0 & -0.50 & -0.19\ m \\ -0.50 & 0 & 0.87 & 0.59\ m \\ 0 & 0 & 0 & 1 \end{pmatrix}$

- ${}^{0}\boldsymbol{H}_2 = \begin{pmatrix} 0.50 & 0.87 & 0 & 0.20\ m \\ -0.75 & 0.43 & -0.50 & -0.21\ m \\ -0.44 & 0.25 & 0.87 & 0.56\ m \\ 0 & 0 & 0 & 1 \end{pmatrix}$

The Hybrid Jacobian Matrices for the two body-fixed coordinate frames Ψ_1 and Ψ_2 for the given joint coordinate set are

- $\boldsymbol{J}^1(q) =$

$$\left(\begin{matrix} 1 & 0 & 0.19\ \frac{m}{rad} & 0.19\ \frac{m}{rad} \\ 0 & 1 & 0.17\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ 0 & 0 & 0\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 0 & 0 \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 0 & 0 \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 1 & 1 \end{matrix}\right.$$

$$\left.\begin{matrix} 0\ \frac{m}{rad} & 0\ \frac{m}{rad} & 0\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ -0.35\ \frac{m}{rad} & -0.27\ \frac{m}{rad} & -0.15\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ -0.16\ \frac{m}{rad} & -0.02\ \frac{m}{rad} & -0.09\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -0.50 \\ 0 & 0 & 0 & 0.87 \end{matrix}\right)$$

- $\boldsymbol{J}^2(q) =$

$$\left(\begin{matrix} 1 & 0 & 0.21\ \frac{m}{rad} & 0.21\ \frac{m}{rad} \\ 0 & 1.00 & 0.20\ \frac{m}{rad} & 0.04\ \frac{m}{rad} \\ 0 & 0 & 0\ \frac{m}{rad} & 0\ \frac{m}{rad} \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 0 & 0 \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 0 & 0 \\ 0\ \frac{rad}{m} & 0\ \frac{rad}{m} & 1 & 1 \end{matrix}\right.$$

$$\left.\begin{matrix} 0\ \frac{m}{rad} & 0\ \frac{m}{rad} & 0\ \frac{m}{rad} & 0.04\ \frac{m}{rad} \\ -0.32\ \frac{m}{rad} & -0.24\ \frac{m}{rad} & -0.12\ \frac{m}{rad} & 0.03\ \frac{m}{rad} \\ -0.18\ \frac{m}{rad} & -0.04\ \frac{m}{rad} & -0.11\ \frac{m}{rad} & 0.02\ \frac{m}{rad} \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -0.50 \\ 0 & 0 & 0 & 0.87 \end{matrix}\right)$$

The mass matrix in the given configuration is:

$$\boldsymbol{M}(\boldsymbol{q}) = \left(\begin{matrix}
31.57\ kg & 0\ kg & 0.54\ kgm & 0.54\ kgm \\
0\ kg & 31.57\ kg & 1.06\ kgm & 0.03\ kgm \\
0.54\ kgm & 1.06\ kgm & 1.27\ \frac{kgm^2}{rad} & 0.09\ \frac{kgm^2}{rad} \\
0.54\ kgm & 0.03\ kgm & 0.09\ \frac{kgm^2}{rad} & 0.08\ \frac{kgm^2}{rad} \\
0\ kgm & -0.70\ kgm & -0.12\ \frac{kgm^2}{rad} & 6.44\text{e-}5\ \frac{kgm^2}{rad} \\
0\ kgm & -0.41\ kgm & -0.07\ \frac{kgm^2}{rad} & 0.14\text{e-}2\ \frac{kgm^2}{rad} \\
0\ kgm & -0.14\ kgm & -0.02\ \frac{kgm^2}{rad} & 7.82\text{e-}4\ \frac{kgm^2}{rad} \\
0\ kgm & 7.14\text{e-}4\ kgm & 3.59\text{e-}4\ \frac{kgm^2}{rad} & 2.39\text{e-}4\ \frac{kgm^2}{rad}
\end{matrix}\right.$$

$$\left.\begin{matrix}
0\ kgm & 0\ kgm & 0\ kgm & 0\ kgm \\
-0.70\ kgm & -0.41\ kgm & -0.14\ kgm & 7.14\text{e-}4\ kgm \\
-0.12\ \frac{kgm^2}{rad} & -0.07\ \frac{kgm^2}{rad} & -0.02\ \frac{kgm^2}{rad} & 3.59\text{e-}4\ \frac{kgm^2}{rad} \\
6.44\text{e-}5\ \frac{kgm^2}{rad} & 0.14\text{e-}2\ \frac{kgm^2}{rad} & 7.82\text{e-}4\ \frac{kgm^2}{rad} & 2.39\text{e-}4\ \frac{kgm^2}{rad} \\
0.20\ \frac{kgm^2}{rad} & 0.10\ \frac{kgm^2}{rad} & 0.05\ \frac{kgm^2}{rad} & -2.76\text{e-}4\ \frac{kgm^2}{rad} \\
0.10\ \frac{kgm^2}{rad} & 0.08\ \frac{kgm^2}{rad} & 0.03\ \frac{kgm^2}{rad} & -1.65\text{e-}4\ \frac{kgm^2}{rad} \\
0.05\ \frac{kgm^2}{rad} & 0.03\ \frac{kgm^2}{rad} & 0.02\ \frac{kgm^2}{rad} & -1.09\text{e-}4\ \frac{kgm^2}{rad} \\
-2.76\text{e-}4\ \frac{kgm^2}{rad} & -1.65\text{e-}4\ \frac{kgm^2}{rad} & -1.09\text{e-}4\ \frac{kgm^2}{rad} & 2.76\text{e-}4\ \frac{kgm^2}{rad}
\end{matrix}\right)$$

The matrix $\boldsymbol{\Lambda}_\epsilon^{-1}$ for body-fixed coordinate frames Ψ_1 and Ψ_2 is:

$$\boldsymbol{\Lambda}_{\epsilon,\Psi_1}^{-1} = \boldsymbol{\Lambda}_{\epsilon,\Psi_2}^{-1} = \begin{pmatrix}
92.80\ \frac{rad}{kgm^2} & -11.27\ \frac{rad}{kgm^2} & 18.21\ \frac{rad}{kgm^2} \\
-11.27\ \frac{rad}{kgm^2} & 909.38\ \frac{rad}{kgm^2} & -1568.90\ \frac{rad}{kgm^2} \\
18.21\ \frac{rad}{kgm^2} & -1568.90\ \frac{rad}{kgm^2} & 2720.70\ \frac{rad}{kgm^2}
\end{pmatrix}.$$

The respective Eigenvalues and Eigenvectors of $\boldsymbol{\Lambda}_\epsilon^{-1}$ are the same:

- Eigenvalues: $\lambda_1 = 0.37$, $\lambda_2 = 9.54\text{e-}4$, $\lambda_3 = 1.34\text{e-}6$
- (Right) Eigenvectors: $u_1 = \begin{pmatrix} -0.01 \\ 0.50 \\ -0.87 \end{pmatrix}, u_2 = \begin{pmatrix} -1.00 \\ 0.00 \\ 0.01 \end{pmatrix}, u_3 = \begin{pmatrix} 0.01 \\ 0.86 \\ 0.50 \end{pmatrix}$

APPENDIX B

Lie algebra structure of translations and rotations

The Lie algebra $se(3)$ can be decomposed into the semi-direct product of a reducable sub-algebra $\mathfrak{t}$ and a simple sub-algebra $so(3)$ (Levi, 1905):

$$se(3) = \mathfrak{t} \rtimes so(3). \tag{B.1}$$

Hereby, the sub-algebra $\mathfrak{t}$ and the sub-algebra $so(3)$ can be associated to translational motion $\boldsymbol{v}$ and rotational motion $\boldsymbol{w}$, respectively. Both sub-algebras are equipped with a bracket operation.

Definition B.1 *For a given $\mathbb{R}$-vector space, a Lie algebra L is equipped with a Lie bracket $[\bullet, \bullet]$, which satisfies three conditions:*

1. *Bi-linearity:* $[\bullet, \bullet] : \forall L \in \Gamma(\mathbb{R}^n) : L \times L \to L$
2. *Anti-symmetry:* $\forall x, y \in L : [x, y] = -[y, x]$
3. *Jacobi identity:* $\forall x, y, z \in L : [x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$

For twists with only angular components $\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_2 \in so(3)$, the bracket operation can be associated with the cross product in $\mathbb{R}^3$:

$$[\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_2] = \tilde{\boldsymbol{w}}_1\tilde{\boldsymbol{w}}_2 - \tilde{\boldsymbol{w}}_2\tilde{\boldsymbol{w}}_1 = (\boldsymbol{w}_1 \times \boldsymbol{w}_2)^{\sim}. \tag{B.2}$$

Hereby, the operation $[\tilde{\ }]$ converts a vector to a skew-symmetric matrix. The bracket operation of $se(3)$ represents a generalization of the cross product on $\mathbb{R}^3$ for twists $\tilde{\boldsymbol{\xi}}_1, \tilde{\boldsymbol{\xi}}_2 \in se(3)$ in $\mathbb{R}^6$:

$$[\tilde{\boldsymbol{\xi}}_1, \tilde{\boldsymbol{\xi}}_2] = \begin{pmatrix} (\boldsymbol{w}_1 \times \boldsymbol{w}_2)^{\sim} & \boldsymbol{w}_1 \times \boldsymbol{v}_2 - \boldsymbol{w}_2 \times \boldsymbol{v}_1 \\ 0 & 0 \end{pmatrix}. \tag{B.3}$$

The sub-algebra $\mathfrak{t}$ is called “an ideal”:

$$[\mathfrak{t}, \bullet] \subseteq \mathfrak{t}. \tag{B.4}$$

In practice this means that any input for the second slot of $[\mathfrak{t}, \bullet]$ will again yield a translational motion, e.g.,

$$[\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{\xi}}_2] = \begin{pmatrix} 0 & \boldsymbol{w}_2 \times \boldsymbol{v}_{\mathbf{1}} \\ 0 & 0 \end{pmatrix} \subseteq \mathfrak{t}. \tag{B.5}$$

The sub-algebra $so(3)$ is called simple and satisfies

$$[so(3), so(3)] = so(3). \tag{B.6}$$

This can be seen in eq. (B.2). With the examples of eqs. (B.3) and (B.5), the expression “semi-direct product” (eq. B.1) can be explained: If rotations are involved, the rotational part of one twist always acts on the translational counterpart of the other twist.

APPENDIX C

Tensor Geometry

C.1 Tensor definition

Consider a point $\boldsymbol{q}$ on a n-dimensional differentiable manifold Q. For all curves on this manifold going through point $\boldsymbol{q}$, a vector space $\mathscr{V}$ exists that collects all *contravariant elements* tangent to the curves. In the point $\boldsymbol{q}$ also a dual vector space $\mathscr{V}^*$ exists, collecting all *covariant elements*.

C.1.1 Covariant tensors

A covariant tensor is a multilinear function of type:

$$\boldsymbol{Q} : \underbrace{\mathscr{V} \times ... \times \mathscr{V}}_{s\ many} \rightarrow \mathbb{R}. \tag{C.1}$$

The function $\boldsymbol{Q}$ takes s many elements of $\mathscr{V}$ and returns a real number. A covariant tensor with $s = 1$ is called a co-vector.

C.1.2 Contravariant tensors

A contravariant tensor is a multilinear function on a covariant tensor:

$$\boldsymbol{P} : \underbrace{\mathscr{V}^* \times ... \times \mathscr{V}^*}_{r\ many} \rightarrow \mathbb{R}. \tag{C.2}$$

The function $\boldsymbol{P}$ takes r many elements of $\mathscr{V}^*$ and returns a real number. A contravariant tensor with $r = 1$ is called a vector.

C.1.3 Mixed tensors

Mixed tensors are r-times contravariant and s-times covariant:

$$\boldsymbol{T} : \underbrace{\mathscr{V}^* \times ... \times \mathscr{V}^*}_{r\ many} \times \underbrace{\mathscr{V} \times ... \times \mathscr{V}}_{s\ many} \rightarrow \mathbb{R}. \tag{C.3}$$

For function $\boldsymbol{T}$, the notation $\binom{r}{s}$ can be used. The number $(r+s)$ defines the rank of the tensor. The tensor can be denoted with r many contravariant superscripts and s many covariant subscripts. For example, a $\binom{1}{1}$-tensor with rank 2 has one contravariant superscript i and one covariant superscript j: $\boldsymbol{T}^i_j$.

C.2 Example of tensor contraction

Operations on tensors can be illustrated with an example:

$$\boldsymbol{\eta_k} = \boldsymbol{g_{ik}} \boldsymbol{\xi^i}, \tag{C.4}$$

with $i = k = 1, ..., n$. On the right side of eq. (C.4), a contravariant tensor $\boldsymbol{\xi^i}$ and a twice covariant tensor $\boldsymbol{g_{ik}}$ appears. The left side yields a covariant tensor $\boldsymbol{\eta_k}$. The operation on the appearing tensors can be seen: The tensor $\boldsymbol{g_{ik}}$ is used to transform the contravariant tensor $\boldsymbol{\xi^i}$ to a covariant tensor $\boldsymbol{\eta_k}$. This operation on tensors is called “contraction.” Contraction sums pairs of equal indices, which are one upper (contravariant) index and one lower (covariant) index:

$$\boldsymbol{\eta_k} = \sum_{i=1}^{n} \boldsymbol{g_{ik}} \boldsymbol{\xi^i}. \tag{C.5}$$

As a result, the index can be canceled out. Implicitly, the sum symbol is left out.

APPENDIX D

Impedance control based on potential energy function

The potential energy function $\mathscr{U} : SE(3) \rightarrow \mathbb{R}$ is the sum of the translational potential $\mathscr{U}_\mathrm{p}$ and the rotational potential $\mathscr{U}_\epsilon$. The time derivative of $\mathscr{U}_\mathrm{p}$ and $\mathscr{U}_\epsilon$ are calculated separately. The resulting differential equations have the form $\boldsymbol{F}^T \boldsymbol{\xi}$. Hence, the wrench part of the differential equations can be extracted. Since the control wrench $\boldsymbol{F}^{\mathrm{imp}}$ is supposed to minimize the respective potential energy function:

$$\boldsymbol{F}^{\mathrm{imp}} = \boldsymbol{F}_\mathrm{p}^{\mathrm{imp}} + \boldsymbol{F}_\epsilon^{\mathrm{imp}} = -\boldsymbol{F}, \tag{D.1}$$

where $\boldsymbol{F}_\mathrm{p}^{\mathrm{imp}} \in se^\star(3)$ is deduced from the $\mathscr{U}_\mathrm{p}$ and $\boldsymbol{F}_\epsilon^{\mathrm{imp}} \in se^\star(3)$ is deduced from $\mathscr{U}_\epsilon$. In the following, all entities are expressed in body-fixed coordinates "b." The equilibrium pose is denoted “e” and the stationary inertial coordinate frame is denoted “0.” The following convention is used: A (co-)vector ${}^0\boldsymbol{a}_\mathrm{e}$ is described in coordinate frame “e,” relative to coordinate frame “0.”

D.1 Translational potential energy function

For a given diagonal or symmetric stiffness matrix $\boldsymbol{K}_\mathrm{p} \in \mathbb{R}^{3\times3}$, the potential energy function can be formulated as:

$$\mathscr{U}_\mathrm{p} = \frac{1}{2} \Delta \boldsymbol{p}^T \boldsymbol{K}_\mathrm{p} \, \Delta \boldsymbol{p}, \tag{D.2}$$

with $\Delta \boldsymbol{p} = ({}^0\boldsymbol{p}_\mathrm{b} - \ {}^0\boldsymbol{p}_\mathrm{e})$. Time differentiation of $\mathscr{U}_\mathrm{p}$ yields:

$$\dot{\mathscr{U}}_\mathrm{p} = \Delta \boldsymbol{p}^T \boldsymbol{K}_\mathrm{p} ({}^0\dot{\boldsymbol{p}}_\mathrm{b} -^0 \dot{\boldsymbol{p}}_\mathrm{e}). \tag{D.3}$$

For a constant equilibrium pose and noting that ${}^0\dot{\boldsymbol{p}}_\mathrm{b}$ has a purely linear component ${}^0\boldsymbol{v}_\mathrm{b} \in \mathbb{R}^3$ and a rotational component $({}^0\boldsymbol{w}_\mathrm{b} \times \Delta \boldsymbol{p}) \in \mathbb{R}^3$:

$$\dot{\mathscr{U}}_\mathrm{p} = \Delta \boldsymbol{p}^T \boldsymbol{K}_\mathrm{p} ({}^0\boldsymbol{v}_\mathrm{b} - ({}^0\boldsymbol{w}_\mathrm{b} \times \Delta \boldsymbol{p})). \tag{D.4}$$

Recalling that the cross product can be represented by a matrix multiplication and using the skew-symmetric property $-{}^0\tilde{\boldsymbol{w}}_\mathrm{b}\ \Delta\boldsymbol{p} =\ \Delta\tilde{\boldsymbol{p}}\ {}^0\boldsymbol{w}_\mathrm{b}$:

$$\dot{\mathscr{U}}_\mathrm{p} = \Delta\boldsymbol{p}^T\boldsymbol{K}_\mathrm{p}({}^0\boldsymbol{v}_\mathrm{b} +\ \Delta\tilde{\boldsymbol{p}}\ {}^0\boldsymbol{w}_\mathrm{b}). \tag{D.5}$$

This equation can be expressed in vector form:

$$\dot{\mathscr{U}}_\mathrm{p} = \underbrace{\begin{pmatrix} \Delta\boldsymbol{p}^T\boldsymbol{K}_\mathrm{p} \\ \Delta\boldsymbol{p}^T\boldsymbol{K}_\mathrm{p}\ \Delta\tilde{\boldsymbol{p}} \end{pmatrix}^T}_{(-\boldsymbol{F}_\mathrm{p}^\mathrm{imp})^T} \underbrace{\begin{pmatrix} {}^0\boldsymbol{v}_\mathrm{b} \\ {}^0\boldsymbol{w}_\mathrm{b} \end{pmatrix}}_{\boldsymbol{\xi}}. \tag{D.6}$$

By using the property $(\Delta\tilde{\boldsymbol{p}})^T = -\Delta\tilde{\boldsymbol{p}}$ and for a diagonal or symmetric matrix $\boldsymbol{K}_\mathrm{p}$, it can be concluded:

$$\boldsymbol{F}_\mathrm{p}^\mathrm{imp} = \begin{pmatrix} -\boldsymbol{K}_\mathrm{p}\ \Delta\boldsymbol{p} \\ \Delta\tilde{\boldsymbol{p}}\ \boldsymbol{K}_\mathrm{p}\ \Delta\boldsymbol{p} \end{pmatrix}. \tag{D.7}$$

D.2 Rotational potential energy function

To derive the rotational potential energy function, the rotation matrix ${}^\mathrm{b}\boldsymbol{R}_\mathrm{e} \in SO(3)$ is converted to a unit-length quaternion representation with unit-axis ${}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e} \in \mathbb{R}^3$ and rotation angle $\eta \in \mathbb{R}$. For a given diagonal or symmetric stiffness matrix $\boldsymbol{K}_\epsilon \in \mathbb{R}^{3\times3}$, the rotational potential energy function can be formulated as:

$$\mathscr{U}_\epsilon = 2\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}{}^T\boldsymbol{K}_\epsilon\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}. \tag{D.8}$$

Time differentiation of $\mathscr{U}_\epsilon$ yields:

$$\dot{\mathscr{U}}_\epsilon = 4\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}{}^T\boldsymbol{K}_\epsilon\ {}^\mathrm{b}\dot{\boldsymbol{\epsilon}}_\mathrm{e}. \tag{D.9}$$

Taking into account the derivation provided in Khalil and Dombre (2002):

$${}^\mathrm{b}\dot{\boldsymbol{\epsilon}}_\mathrm{e} = -\frac{1}{2}\boldsymbol{E}(\eta,{}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e})\ {}^\mathrm{e}\boldsymbol{w}_\mathrm{b}, \tag{D.10}$$

with

$$\boldsymbol{E}(\eta,{}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}) = \eta\boldsymbol{I} -{}^\mathrm{b}\tilde{\boldsymbol{\epsilon}}_\mathrm{e}. \tag{D.11}$$

Substituting eq. (D.10) in eq. (D.9) yields:

$$\dot{\mathscr{U}}_\epsilon = -2\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}{}^T\boldsymbol{K}_\epsilon\ \boldsymbol{E}(\eta,{}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e})\ {}^\mathrm{e}\boldsymbol{w}_\mathrm{b}. \tag{D.12}$$

This equation can be expressed in vector form:

$$\dot{\mathscr{U}}_\epsilon = \underbrace{\begin{pmatrix} 0 \\ -2\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}{}^T\boldsymbol{K}_\epsilon\ \boldsymbol{E}(\eta,{}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e}) \end{pmatrix}^T}_{(-\boldsymbol{F}_\epsilon^\mathrm{imp})^T} \underbrace{\begin{pmatrix} {}^\mathrm{e}\boldsymbol{v}_\mathrm{b} \\ {}^\mathrm{e}\boldsymbol{w}_\mathrm{b} \end{pmatrix}}_{\boldsymbol{\xi}}. \tag{D.13}$$

For a diagonal or symmetric matrix $\boldsymbol{K}_\epsilon$, it can be concluded:

$$\boldsymbol{F}_\epsilon^\mathrm{imp} = \begin{pmatrix} 0 \\ 2\ \boldsymbol{E}(\eta,{}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e})^T\ \boldsymbol{K}_\epsilon\ {}^\mathrm{b}\boldsymbol{\epsilon}_\mathrm{e} \end{pmatrix}. \tag{D.14}$$

D.3 Time differentiation of energy functions with time-varying stiffness

D.3.1 Translational potential energy function

As shown in eq. (3.49a), the translational potential energy function with time-varying components $\Delta\boldsymbol{p}$, $\bar{\boldsymbol{\epsilon}}$, $\bar{\eta}$, and $\bar{\boldsymbol{k}}$ can be represented by:

$$\mathscr{U}_{\mathrm{p}}(\Delta\boldsymbol{p}, \bar{\boldsymbol{\epsilon}}, \bar{\eta}, \bar{\boldsymbol{k}}) = \frac{1}{2}\Delta\boldsymbol{p}^T \exp(\tilde{\bar{\boldsymbol{\epsilon}}}\bar{\eta})\ \mathrm{diag}(\bar{\boldsymbol{k}})\ \exp(\tilde{\bar{\boldsymbol{\epsilon}}}^T\bar{\eta})\ \Delta\boldsymbol{p}. \tag{D.15}$$

Time differentiation yields:

$$\dot{\mathscr{U}}_{\mathrm{p}} = (\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Delta\boldsymbol{p}})^T \dot{\Delta\boldsymbol{p}} + (\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{k}}})^T \dot{\bar{\boldsymbol{k}}} + (\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{\epsilon}}})^T \dot{\bar{\boldsymbol{\epsilon}}} + (\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\eta}})^T \dot{\bar{\eta}}. \tag{D.16}$$

Note that $\dot{\boldsymbol{\epsilon}}$ was defined in eq. (D.10) and can be found in Khalil and Dombre (2002). In the following, the individual summands of eq. (D.16) will be derived. The first summand was shown in eq. (D.6):

$$(\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Delta\boldsymbol{p}})^T \dot{\Delta\boldsymbol{p}} = (-\boldsymbol{F}_{\mathrm{p}}^{\mathrm{imp}})^T \boldsymbol{\xi}. \tag{D.17}$$

For the second summand, $\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{k}}} = [\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{k}_1}, \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{k}_2}, \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{k}_3}]^T$, which can be expanded to $\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{k}}} = [\Delta\boldsymbol{p}^T\boldsymbol{r}_1\boldsymbol{r}_1^T\Delta\boldsymbol{p}, \Delta\boldsymbol{p}^T\boldsymbol{r}_2\boldsymbol{r}_2^T\Delta\boldsymbol{p}, \Delta\boldsymbol{p}^T\boldsymbol{r}_3\boldsymbol{r}_3^T\Delta\boldsymbol{p}]^T$. Here, $\boldsymbol{r}_i$ is the i-th column of $\boldsymbol{R} \in SO(3)$ and $\bar{k}_i \in \mathbb{R}$ is the i-th element of vector $\bar{\boldsymbol{k}}$. Hence, the second summand can be written in matrix notation:

$$(\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{k}}})^T \dot{\bar{\boldsymbol{k}}} = \frac{1}{2}\Delta\boldsymbol{p}^T\boldsymbol{R}\dot{\bar{\boldsymbol{K}}}\boldsymbol{R}^T\Delta\boldsymbol{p}. \tag{D.18}$$

For the third summand, Rodrigues' formula will be needed (eq. (3.48)). Note that $\tilde{\bar{\boldsymbol{\epsilon}}}^2 = \begin{pmatrix} \epsilon_2^2+\epsilon_3^2 & \epsilon_1\epsilon_2 & \epsilon_1\epsilon_3 \\ \epsilon_1\epsilon_2 & \epsilon_1^2+\epsilon_3^2 & \epsilon_2\epsilon_3 \\ \epsilon_1\epsilon_3 & \epsilon_2\epsilon_3 & \epsilon_1^2+\epsilon_2^2 \end{pmatrix}$. Reforming eq. (D.15) yields:

$$\mathscr{U}_{\mathrm{p}}(\Delta\boldsymbol{p}, \bar{\boldsymbol{\epsilon}}, \bar{\eta}, \bar{\boldsymbol{k}}) = \frac{1}{2}\bar{\boldsymbol{k}}^T(\square_1\square_1^T, \square_2\square_2^T, \square_3\square_3^T)^T, \tag{D.19}$$

where $\square_i = \left(\boldsymbol{I}_i + \tilde{\bar{\boldsymbol{\epsilon}}}_i\sin(\eta) + (\tilde{\bar{\boldsymbol{\epsilon}}}^2)_i(1-\cos(\eta))\right)^T\Delta\boldsymbol{p}$. Here, $\boldsymbol{I}_i$, $\tilde{\bar{\boldsymbol{\epsilon}}}_i$, and $(\tilde{\bar{\boldsymbol{\epsilon}}}^2)_i$ denote the i-th column of the identity matrix, of the matrix $\tilde{\bar{\boldsymbol{\epsilon}}}$, and of the matrix $\tilde{\bar{\boldsymbol{\epsilon}}}^2$, respectively. Using the chain rule and considering that $\bar{\boldsymbol{\epsilon}} = [\epsilon_1, \epsilon_2, \epsilon_3]^T$ yields:

$$\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\boldsymbol{\epsilon}}} = \begin{pmatrix} \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_1}\frac{\partial \square_1}{\partial \epsilon_1} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_2}\frac{\partial \square_2}{\partial \epsilon_1} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_3}\frac{\partial \square_3}{\partial \epsilon_1} \\ \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_1}\frac{\partial \square_1}{\partial \epsilon_2} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_2}\frac{\partial \square_2}{\partial \epsilon_2} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_3}\frac{\partial \square_3}{\partial \epsilon_2} \\ \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_1}\frac{\partial \square_1}{\partial \epsilon_3} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_2}\frac{\partial \square_2}{\partial \epsilon_3} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \square_3}\frac{\partial \square_3}{\partial \epsilon_3} \end{pmatrix}. \tag{D.20}$$

The matrix elements in eq.(D.20) can be derived as follows:

- $\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Box_i} = \bar{\boldsymbol{k}}_i \ \Box_i;$
- $\frac{\partial \Box_1}{\partial \epsilon_1} = (1-\cos(\eta)) \ [0, \epsilon_2, \epsilon_3] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_2}{\partial \epsilon_1} = [\epsilon_2(1-\cos(\eta)), 2\epsilon_1(1-\cos(\eta)), \sin(\eta)] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_3}{\partial \epsilon_1} = [\epsilon_3(1-\cos(\eta)), -\sin(\eta), 2\epsilon_1(1-\cos(\eta))] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_1}{\partial \epsilon_2} = [2\epsilon_2(1-\cos(\eta)), \epsilon_1(1-\cos(\eta)), -\sin(\eta)] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_2}{\partial \epsilon_2} = (1-\cos(\eta))[\epsilon_1, 0, \epsilon_3] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_3}{\partial \epsilon_2} = [\sin(\eta), \epsilon_3(1-\cos(\eta)), 2\epsilon_2(1-\cos(\eta))] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_1}{\partial \epsilon_3} = [2\epsilon_3(1-\cos(\eta)), \sin(\eta), \epsilon_1(1-\cos(\eta))] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_2}{\partial \epsilon_3} = [-\sin(\eta), 2\epsilon_2(1-\cos(\eta)), \epsilon_2(1-\cos(\eta))] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_3}{\partial \epsilon_3} = (1-\cos(\eta))[\epsilon_1, \epsilon_2, 0] \ \Delta \boldsymbol{p}.$

For the fourth summand, again the chain rule has to applied:

$$\frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \bar{\eta}} = \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Box_1}\frac{\partial \Box_1}{\partial \eta} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Box_2}\frac{\partial \Box_2}{\partial \eta} + \frac{\partial \mathscr{U}_{\mathrm{p}}}{\partial \Box_3}\frac{\partial \Box_3}{\partial \eta}. \tag{D.21}$$

The elements $\frac{\partial \Box_i}{\partial \eta}$ can be derived as follows:

- $\frac{\partial \Box_1}{\partial \eta} = [(\epsilon_2^2+\epsilon_3^2)\sin(\eta), \epsilon_1\epsilon_2\sin(\eta)+\epsilon_3\cos(\eta), \epsilon_1\epsilon_3\sin(\eta)-\epsilon_2\cos(\eta)] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_2}{\partial \eta} = [\epsilon_1\epsilon_2\sin(\eta)-\epsilon_3\cos(\eta), (\epsilon_1^2+\epsilon_3^2)\sin(\eta), \epsilon_2\epsilon_3\sin(\eta)+\epsilon_1\cos(\eta)] \ \Delta \boldsymbol{p};$
- $\frac{\partial \Box_3}{\partial \eta} = [\epsilon_1\epsilon_3\sin(\eta)+\epsilon_2\cos(\eta), \epsilon_2\epsilon_3\sin(\eta)-\epsilon_1\cos(\eta), (\epsilon_1^2+\epsilon_2^2)\sin(\eta)] \ \Delta \boldsymbol{p}.$

D.3.2 Rotational potential energy function

As shown in eq. (3.49b), the rotational potential energy function with time-varying components $\boldsymbol{\epsilon}$, $\bar{\boldsymbol{\epsilon}}$, $\bar{\eta}$, and $\bar{\boldsymbol{k}}$ can be represented by:

$$\mathscr{U}_\epsilon(\boldsymbol{\epsilon}, \bar{\boldsymbol{\epsilon}}, \bar{\eta}, \bar{\boldsymbol{k}}) = 2\ \boldsymbol{\epsilon}^T \exp(\tilde{\bar{\boldsymbol{\epsilon}}}\bar{\eta})\ \mathrm{diag}(\bar{\boldsymbol{k}})\ \exp(\tilde{\bar{\boldsymbol{\epsilon}}}^T\bar{\eta})\ \boldsymbol{\epsilon}. \tag{D.22}$$

Time differentiation yields:

$$\dot{\mathscr{U}}_\epsilon = (\frac{\partial \mathscr{U}_\epsilon}{\partial \boldsymbol{\epsilon}})^T \dot{\boldsymbol{\epsilon}} + (\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\boldsymbol{k}}})^T \dot{\bar{\boldsymbol{k}}} + (\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\boldsymbol{\epsilon}}})^T \dot{\bar{\boldsymbol{\epsilon}}} + (\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\eta}})^T \dot{\bar{\eta}}. \tag{D.23}$$

The first summand of eq. (D.23) was shown in eq. (D.13):

$$(\frac{\partial \mathscr{U}_\epsilon}{\partial \boldsymbol{\epsilon}})^T \dot{\boldsymbol{\epsilon}} = (-\boldsymbol{F}_\epsilon^{\mathrm{imp}})^T \boldsymbol{\xi}. \tag{D.24}$$

The second summand can be derived similar to the translational potential energy function:

$$(\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\boldsymbol{k}}})^T \dot{\bar{\boldsymbol{k}}} = 2\ \boldsymbol{\epsilon}^T \boldsymbol{R}\dot{\bar{\boldsymbol{K}}}\boldsymbol{R}^T \boldsymbol{\epsilon}. \tag{D.25}$$

For the third summand, again Rodrigues' formula can be applied. Reforming eq. (D.22) yields:

$$\mathscr{U}_\epsilon(\boldsymbol{\epsilon}, \bar{\boldsymbol{\epsilon}}, \bar{\eta}, \bar{\boldsymbol{k}}) = 2\ \bar{\boldsymbol{k}}^T (\boxdot_1 \boxdot_1^T, \boxdot_2 \boxdot_2^T, \boxdot_3 \boxdot_3^T)^T, \tag{D.26}$$

where $\boxdot_i = \left(\boldsymbol{I}_i + \tilde{\bar{\boldsymbol{\epsilon}}}_i \sin(\eta) + (\tilde{\bar{\boldsymbol{\epsilon}}}^2)_i (1-\cos(\eta))\right)^T \boldsymbol{\epsilon}$. Again, $\boldsymbol{I}_i$, $\tilde{\bar{\boldsymbol{\epsilon}}}_i$, and $(\tilde{\bar{\boldsymbol{\epsilon}}}^2)_i$ denote the i-th column of the identity matrix, of the matrix $\tilde{\bar{\boldsymbol{\epsilon}}}$, and of the matrix $\tilde{\bar{\boldsymbol{\epsilon}}}^2$, respectively. Using the chain rule and considering that $\bar{\boldsymbol{\epsilon}} = [\epsilon_1, \epsilon_2, \epsilon_3]^T$,

$$\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\boldsymbol{\epsilon}}} = \begin{pmatrix} \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_1}\frac{\partial \boxdot_1}{\partial \epsilon_1} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_2}\frac{\partial \boxdot_2}{\partial \epsilon_1} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_3}\frac{\partial \boxdot_3}{\partial \epsilon_1} \\ \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_1}\frac{\partial \boxdot_1}{\partial \epsilon_2} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_2}\frac{\partial \boxdot_2}{\partial \epsilon_2} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_3}\frac{\partial \boxdot_3}{\partial \epsilon_2} \\ \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_1}\frac{\partial \boxdot_1}{\partial \epsilon_3} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_2}\frac{\partial \boxdot_2}{\partial \epsilon_3} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_3}\frac{\partial \boxdot_3}{\partial \epsilon_3} \end{pmatrix}. \tag{D.27}$$

The matrix elements in eq.(D.27) can be derived as follows:

- $\dfrac{\partial \mathscr{U}_\mathrm{p}}{\partial \boxdot_i} = 4\ \bar{\boldsymbol{k}}_i\ \boxdot_i$;
- $\dfrac{\partial \boxdot_1}{\partial \epsilon_1} = (1-\cos(\eta))\ [0, \epsilon_2, \epsilon_3]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_2}{\partial \epsilon_1} = [\epsilon_2(1-\cos(\eta)), 2\epsilon_1(1-\cos(\eta)), \sin(\eta)]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_3}{\partial \epsilon_1} = [\epsilon_3(1-\cos(\eta)), -\sin(\eta), 2\epsilon_1(1-\cos(\eta))]\ \boldsymbol{\epsilon}$;

- $\dfrac{\partial \boxdot_1}{\partial \epsilon_2} = [2\epsilon_2(1-\cos(\eta)), \epsilon_1(1-\cos(\eta)), -\sin(\eta)]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_2}{\partial \epsilon_2} = (1-\cos(\eta))[\epsilon_1, 0, \epsilon_3]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_3}{\partial \epsilon_2} = [\sin(\eta), \epsilon_3(1-\cos(\eta)), 2\epsilon_2(1-\cos(\eta))]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_1}{\partial \epsilon_3} = [2\epsilon_3(1-\cos(\eta)), \sin(\eta), \epsilon_1(1-\cos(\eta))]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_2}{\partial \epsilon_3} = [-\sin(\eta), 2\epsilon_2(1-\cos(\eta)), \epsilon_2(1-\cos(\eta))]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_3}{\partial \epsilon_3} = (1-\cos(\eta))[\epsilon_1, \epsilon_2, 0]\ \boldsymbol{\epsilon}$.

Similar, for the fourth summand:

$$\frac{\partial \mathscr{U}_\epsilon}{\partial \bar{\eta}} = \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_1}\frac{\partial \boxdot_1}{\partial \eta} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_2}\frac{\partial \boxdot_2}{\partial \eta} + \frac{\partial \mathscr{U}_\epsilon}{\partial \boxdot_3}\frac{\partial \boxdot_3}{\partial \eta}. \tag{D.28}$$

The elements $\dfrac{\partial \boxdot_i}{\partial \eta}$ can be derived as follows:

- $\dfrac{\partial \boxdot_1}{\partial \eta} = [(\epsilon_2^2+\epsilon_3^2)\sin(\eta), \epsilon_1\epsilon_2\sin(\eta)+\epsilon_3\cos(\eta), \epsilon_1\epsilon_3\sin(\eta)-\epsilon_2\cos(\eta)]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_2}{\partial \eta} = [\epsilon_1\epsilon_2\sin(\eta)-\epsilon_3\cos(\eta), (\epsilon_1^2+\epsilon_3^2)\sin(\eta), \epsilon_2\epsilon_3\sin(\eta)+\epsilon_1\cos(\eta)]\ \boldsymbol{\epsilon}$;
- $\dfrac{\partial \boxdot_3}{\partial \eta} = [\epsilon_1\epsilon_3\sin(\eta)+\epsilon_2\cos(\eta), \epsilon_2\epsilon_3\sin(\eta)-\epsilon_1\cos(\eta), (\epsilon_1^2+\epsilon_2^2)\sin(\eta)]\ \boldsymbol{\epsilon}$.

APPENDIX E

Bond Graphs

E.1 Elements

Tab. E.1 shows the Bond Graph elements that are partly used in this work and which can be defined by their essential connection between flow and effort.

Table E.1: Essential connection of Bond Graph elements.

Symbol	**Notation**	**Essential connection**
$\boldsymbol{I}$	inertial element	flow $= 1/\boldsymbol{I}\ \int_{-\infty}^{\text{now}}$ effort dt
$\boldsymbol{K}$	compliant element	effort $= \boldsymbol{K}\ \int_{-\infty}^{\text{now}}$ flow dt
$\boldsymbol{B}$	resistive element	effort $= \boldsymbol{B}$ flow
$\mathcal{S}_f$	source of flow	flow
$\mathcal{S}_e$	source of effort	effort
1	1-junction	effort-summing junction
0	0-junction	flow-summing junction
TF	ideal transformer	flow *and* effort
MTF	modulated transformer	flow *and* effort

E.2 Connection via multi-bonds

For multi-dimensional systems, the energy flow is represented by a double-lined arrow called "power-bond" or "multi-bond." The multi-bond points in the positive direction of energy flow. The creator of the Bond Graph model is free to chose the direction of power. The origin of the flow is indicated by a vertical line attached to the multi-bond.

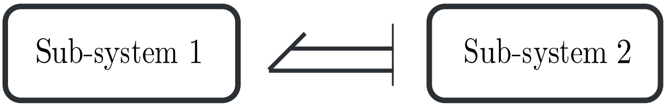


Figure E.1: Multi-bond, indicating the direction of power flow between two sub-systems.

E.3 Causality

Bond graphs can be used to understand the functionality of a physical system since they make the causality of the physical elements explicit.

Table E.2: Causality of Bond Graph elements.

Symbol	**Causality**
$\mathscr{S}_f, \mathscr{S}_e$	produces power
$\mathscr{R}$	absorbs power
$\mathbb{C}, \boldsymbol{I}$	stores power
MTF	power in, power out

APPENDIX F

Shaping Impedances

F.1 Controller parameters

Table F.1: Controller parameters for the experiment 1 in sec. 3.5.1. The scalar stiffnesses $k^i_{p/\epsilon}$ are the elements on the main diagonal of the matrices $\boldsymbol{K}^i_{p/\epsilon}$, respectively.

Experiment 1		
Initial configuration (degrees) $\boldsymbol{q} = [-15.80, 30.00, 24.23, -90.49, 6.48, -60.00, 36.48]$		
Impedances	$k_p^1 = 1500 \ \frac{\mathrm{kg}}{\mathrm{s}^2}$ $\zeta^1 = 0.8$	$k_p^2 = 500 \ \frac{\mathrm{kg}}{\mathrm{s}^2}$ $\zeta^2 = 0.8$
Trajectory	$A = 0.15$ m	$t^{\mathrm{lin}} = 3$ s
Robot mass		$m^{\mathrm{rob}} = 26.334$ kg
Mass ratio		$a = 0.3$
Nullspace dim.		2
Joint damping	$\beta^{\mathrm{ns}} = 1.2 \ \mathrm{kgm}^2/\mathrm{s}$	$\boldsymbol{B}_q = \beta^{\mathrm{ns}} \ \boldsymbol{I}$

Table F.2: Controller parameters for the experiment 2 in sec. 3.5.2. The scalar stiffnesses $k^i_{p/\epsilon}$ are the elements on the main diagonal of the matrices $\boldsymbol{K}^i_{p/\epsilon}$, respectively. Due to instabilities, the impedances of the experiment were reduced for the nullspace projection method.

Experiment 2		
Initial configuration (degrees)		
$\boldsymbol{q} = [-15.64, 53.09, 24.23, -88.97, -10.21, -69.24, 43.93]$		
Superposition and Impedance Shaping		
$k^1_p = 1500\ \frac{\text{kg}}{\text{s}^2}$	$k^2_p = 800\ \frac{\text{kg}}{\text{s}^2}$	$k^3_\epsilon = 60\ \frac{\text{kgm}^2}{\text{s}^2}$
$\zeta^1 = 0.6$	$\zeta^2 = 0.6$	$\zeta^3 = 0.6$
Nullspace projection		
$k^1_p = 1000\ \frac{\text{kg}}{\text{s}^2}$	$k^2_p = 500\ \frac{\text{kg}}{\text{s}^2}$	$k^3_\epsilon = 40\ \frac{\text{kgm}^2}{\text{s}^2}$
$\zeta^1 = 0.6$	$\zeta^2 = 0.6$	$\zeta^3 = 0.6$
Trajectory		
	$D = 0.07$ m	$t^{\text{circ}} = 5$ s
Robot mass		$m^{\text{rob}} = 26.334$ kg
Ratios	$a = 0.3$	$b = 1400$
Nullspace dim.		0
Joint damping		
	$\beta^{\text{ns}} = 1.2\ E\text{kgm}^2/\text{s}$	$\boldsymbol{B}_q = \beta^{\text{ns}}\ \boldsymbol{I}$

APPENDIX G

Energy Budgets

G.1 Set-up for collision measurements

The set-up for the collision measurement is shown in fig. G.1 and all components are listed in tab. G.1.

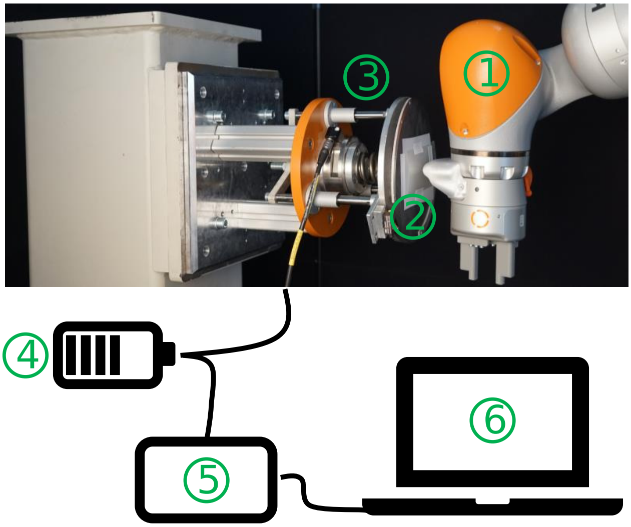


Figure G.1: Set-up for collision measurements

Table G.1: Components of measurement set-up, shown in fig. G.1.

Pos.	Description	Manufacturer
1	LBR iiwa R820	KUKA
2	Pressure sheet "Prescale LLW"	Fujifilm
3	Collision measurement device FT Delta sensor FT09023 Pressure spring D-380 (hand) and D-339N-03 (arm) Elastomer NK-SH40-10mm Linear bearing LHIRD-12 (3x) Hollow shaft SPJW12-160-M8 (3x)	KUKA Schunk Gutekunst Schippel MISUMI MISUMI
4	Power supply 240VAC-763067	National Instruments
5	Data logger USB-6341 Sensor cable 30053244 FTD-C-H-PS-6	National Instruments National Instruments
6	Laptop Precision 7520 Software LabView (Data acquisition force sensor) Matlab R2017b (Data processing and visualization) Software FDP-8010E (Evaluation contact surface area)	Dell National Instruments Mathworks Fujifilm
	Scanner V37 (Scanning of pressure sheet)	Epson

G.2 Controller parameters

To verify the energy budget calculated with the thresholds of ISO/TS 15066, the controller of sec. 4.3.1 was implemented with initial control parameters given below:

- $\boldsymbol{K}_q = 0\ \frac{\text{kgm}}{\text{s}^2}\boldsymbol{I_7}$, where $\boldsymbol{I_7} \in \mathbb{R}^{7\times 7}$ is the identity matrix $\rightarrow \mathscr{U}(\Delta\boldsymbol{q}) = 0$ J
- $\boldsymbol{K}_p = 1500\ \frac{\text{kg}}{\text{s}^2}\boldsymbol{I_3}$, where $\boldsymbol{I_3} \in \mathbb{R}^{3\times 3}$ is the identity matrix
- $\boldsymbol{K}_\epsilon = 150\ \frac{\text{kgm}^2}{\text{s}^2}\boldsymbol{I_3}$
- $\zeta = 0.8$
- $\beta^{\text{ns}} = 0.5\frac{\text{kgm}}{\text{s}}$

- $\mathscr{L}_{\mathrm{max}} = 0.52$ J
- $\boldsymbol{K}_q{}^{\bullet} = 1\ \frac{\mathrm{kgm}}{\mathrm{s}^2} * \boldsymbol{I_7}$
- $\Omega = 40$

The desired control motion is dependent on t_{eff} with motion along the x-coordinate during the transfer motion and motion along the z-coordinate during the approach motion. Both coordinates are expressed in the inertial coordinate frame 0. The first part of the desired control motion of the example application can be seen in tab. G.2, with parameters below:

- $A_{\mathrm{x}} = 0.7$ m ; $A_{\mathrm{z}} = 0.3$ m
- $t_{\mathrm{x}} = 4$ s ; $t_{\mathrm{z}} = 3$ s
- $x_{\mathrm{init}} = -0.21$ m
- $y_{\mathrm{init}} = 0.56$ m
- $z_{\mathrm{init}} = 0.01$ m

Table G.2: Desired control motion for the first part of the example application.

Time	**Desired control motion**
$t_{\mathrm{eff}} \leq t_{\mathrm{x}}$	${}^0x_{\mathrm{c}} = {}^0x_{\mathrm{init}} - 0.5\ A_{\mathrm{x}}\ (1 - \cos(\pi\ \frac{t_{\mathrm{eff}}}{t_{\mathrm{x}}}))$ ${}^0y_{\mathrm{c}} = {}^0y_{\mathrm{init}}$ ${}^0z_{\mathrm{c}} = {}^0z_{\mathrm{init}}$
$t_{\mathrm{x}} \leq t_{\mathrm{eff}} \leq (t_{\mathrm{x}} + t_{\mathrm{z}})$	${}^0x_{\mathrm{c}} = {}^0x_{\mathrm{c}}$ ${}^0y_{\mathrm{c}} = {}^0y_{\mathrm{init}}$ ${}^0z_{\mathrm{c}} = {}^0z_{\mathrm{init}} - \frac{t_{\mathrm{eff}} - t_{\mathrm{x}}}{t_{\mathrm{z}}}\ A_{\mathrm{z}}$

Bibliography

Ajoudani A, Tsagarakis NG and Bicchi A (2015) On the role of robot configuration in Cartesian stiffness control. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA, pp. 1010–1016. doi: 10.1109/ICRA.2015.7139300.

Albu-Schaffer A, Ott C, Frese U and Hirzinger G (2003) Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. *IEEE International Conference on Robotics and Automation* : 3704–3709.

Allmendinger F (2015) Computational methods for the kinematic analysis of diarthrodial joints.

Angeles J (2014) *Fundamentals of robotic mechanical systems: Theory, methods, and algorithms*. Mechanical engineering series, 4. ed. edition. Cham: Springer. ISBN 978-3-319-01850-8.

Asada H (1983) A Geometrical Representation of Manipulator Dynamics and Its Application to Arm Design. *Journal of Dynamic Systems, Measurement, and Control* 105(3): 131. doi: 10.1115/1.3140644.

Azad M and Babič J (2019) Effects of the weighting matrix on dynamic manipulability of robots. *Autonomous Robots* 43(7): 1867–1879. doi: 10.1007/s10514-018-09819-y.

Baillieul J (1987) A constraint oriented approach to inverse problems for kinematically redundant manipulators. *Proceedings of the 1987 IEEE International Conference on Robotics and Automation* 1: 1827–1833. doi: 10.1109/ROBOT.1987.1087809.

Beetz M, Bartels G, Albu-Schaffer A, Balint-Benczedi F, Belder R, Bebler D, Haddadin S, Maldonado A, Mansfeld N, Wiedemeyer T, Weitschat R and Worch JH (2015) Robotic agents capable of natural and safe physical interaction with human co-workers. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* : 6528–6535.

Bennett DJ, Hollerbach JM, Xu Y and Hunter IW (1992) Time-varying stiffness of human elbow joint during cyclic voluntary movement. *Experimental Brain Research* 88(2): 433–442. doi: 10.1007/BF02259118.

Bergner F, Dean-Leon E, Guadarrama-Olvera JR and Cheng G (2019) Evaluation of a Large Scale Event Driven Robot Skin. *IEEE Robotics and Automation Letters* 4(4): 4247–4254. doi: 10.1109/LRA.2019.2930493.

Bicchi A, Salisbury JK and Brock DL (1993) Contact Sensing from Force Measurements. *The International Journal of Robotics Research* 12(3): 249–262. doi: 10.1177/027836499301200304.

Billard A, Calinon S, Dillmann R and Schaal S (2008) Robot Programming by Demonstration. In: Siciliano B and Khatib O (eds.) *Springer Handbook of Robotics*. Berlin, Heidelberg. ISBN 978-3-540-30301-5, pp. 1371–1394.

Birjandi SAB, Kuhn J and Haddadin S (2020) Observer-Extended Direct Method for Collision Monitoring in Robot Manipulators Using Proprioception and IMU Sensing. *IEEE Robotics and Automation Letters* 5(2): 954–961. doi: 10.1109/LRA.2020.2967287.

Boothby WM (1975) *An introduction to differentiable manifolds and Riemannian geometry*, *Pure and applied mathematics*, volume 63. New York: Academic Press. ISBN 0-12-116050-5.

Buchli J, Stulp F, Theodorou E and Schaal S (2011) Learning variable impedance control. *The International Journal of Robotics Research* 30(7): 820–833. doi: 10.1177/0278364911402527.

Burdet E, Franklin DW and Milner TE (2013) *Human robotics: neuromechanics and motor control*. Cambridge, Massachusetts: The MIT Press. ISBN 978-0-262-01953-8.

Burdet E, Osu R, Franklin DW, Milner TE and Kawato M (2001) The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature* 414(6862): 446–449. doi: 10.1038/35106566.

Burget F and Bennewitz M (2015) Stance selection for humanoid grasping tasks by inverse reachability maps. *Proceedings of the 2015 IEEE International Conference on Robotics and Automation* 1: 5669–5674. doi: 10.1109/ICRA.2015.7139993.

Busson D, Bearee R and Olabi A (2017) Task-oriented rigidity optimization for 7 DOF redundant manipulators. *IFAC-PapersOnLine* 50(1): 14588–14593. doi: 10.1016/j.ifacol.2017.08.2108.

Caccavale F, Natale C, Siciliano B and Villani L (1999) Six-DOF impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation* 15(2): 289–300. doi: 10.1109/70.760350.

Califano F, Rashad R, Dijkshoorn A, Koerkamp LG, Sneep R, Brugnoli A and Stramigioli S (2021) Decoding and realising flapping flight with port-Hamiltonian system theory. *Annual Reviews in Control* 51: 37–46. doi: 10.1016/j.arcontrol.2021.03.009.

Charalampous K, Kostavelis I and Gasteratos A (2017) Recent trends in social aware robot navigation: A survey. *Robotics and Autonomous Systems* 93: 85–104. doi: 10.1016/j.robot.2017.03.002.

Chen JH and Song KT (2018) Collision-Free Motion Planning for Human-Robot Collaborative Safety Under Cartesian Constraint. *2018 IEEE International Conference on Robotics and Automation (ICRA)* : 4348–4354.

Chen Q, Zhu S and Zhang X (2015) Improved Inverse Kinematics Algorithm Using Screw Theory for a Six-DOF Robot Manipulator. *International Journal of Advanced Robotic Systems* 12(10): 140. doi: 10.5772/60834.

Chiu SL (1988) Task Compatibility of Manipulator Postures. *The international journal of robotics research* 1(5): 13–21. doi: 10.1177/027836498800700502.

Colgate J and Hogan N (1988) Robust control of dynamically interacting systems. *International Journal of Control* 48(1): 65–88. doi: 10.1080/00207178808906161.

De Luca A, Albu-Schaffer A, Haddadin S and Hirzinger G (2006) Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* : 1623–1630.

De Luca A and Flacco F (2012) Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics* : 288–295.

De Santis A, Siciliano B, De Luca A and Bicchi A (2008) An atlas of physical human–robot interaction. *Mechanism and Machine Theory* 43(3): 253–270. doi: 10.1016/j.mechmachtheory.2007.03.003.

Demming R and Duffy DJ (2012) *Introduction to the Boost C++ libraries.* Amsterdam, The Netherlands: Datasim Education BV. ISBN 978-94-91028-01-4.

Dietrich A, Ott C and Albu-Schäffer A (2015) An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research* 34(11): 1385–1400. doi: 10.1177/0278364914566516.

Dietrich A, Ott C and Stramigioli S (2016) Passivation of Projection-Based Null Space Compliance Control Via Energy Tanks. *IEEE Robotics and Automation Letters* 1(1): 184–191. doi: 10.1109/LRA.2015.2512937.

Dietrich A, Wu X, Bussmann K, Harder M, Iskandar M, Englsberger J, Ott C and Albu-Schäffer A (2021) Practical consequences of inertia shaping for interaction and tracking in robot control. *Control Engineering Practice* 114: 104875. doi: https://doi.org/10.1016/j.conengprac.2021.104875.

Dietrich A, Wu X, Bussmann K, Ott C, Albu-Schaffer A and Stramigioli S (2017) Passive Hierarchical Impedance Control Via Energy Tanks. *IEEE Robotics and Automation Letters* 2(2): 522–529. doi: 10.1109/LRA.2016.2645504.

Dombrowski U, Stefanak T and Reimer A (2018) Simulation of human-robot collaboration by means of power and force limiting. *Procedia Manufacturing* 17: 134–141. doi: 10.1016/j.promfg.2018.10.028.

Duan J, Gan Y, Cao P and Dai X (2019) The Optimal Solution for Base Frame Installation of Dual-arm Robot. *International Conference on Advanced Robotics and Mechatronics (ICARM)* 4: 546–552. doi: 10.1109/ICARM.2019.8833952.

Dubrovin BA, Fomenko AT and Novikov SP (1984) *Modern Geometry - Methods and Applications*, volume 93. New York: Springer New York. doi: 10.1007/978-1-4684-9946-9.

Duindam V, Macchelli A, Stramigioli S and Bruyninckx H (2009) *Modeling and Control of Complex Physical Systems.* Springer Berlin Heidelberg. doi: 10.1007/978-3-642-03196-0.

Duindam V and Stramigioli S (2004) Port-Based Asymptotic Curve Tracking for Mechanical Systems. *European Journal of Control* 10(5): 411–420. doi: 10.3166/ejc.10.411-420.

Escande A, Mansard N and Wieber PB (2014) Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research* 33(7): 1006–1028. doi: 10.1177/0278364914521306.

Eyssautier F and Eyssautier C (2021) Capsix Robotics. URL `https://capsix-robotics.com/`. Accessed: 2020-11-03.

Fasse ED and Broenink JF (1997) A spatial impedance controller for robotic manipulation. *IEEE Transactions on Robotics and Automation* 13(4): 546–556. doi: 10.1109/70.611315.

Faverjon B and Tournassoud P (1987) A local based approach for path planning of manipulators with a high number of degrees of freedom. In: *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4. Raleigh, NC, USA: Institute of Electrical and Electronics Engineers, pp. 1152–1159. doi: 10.1109/ROBOT.1987.1087982.

Ferraguti F, Bertuletti M, Landi CT, Bonfe M, Fantuzzi C and Secchi C (2020) A Control Barrier Function Approach for Maximizing Performance While Fulfilling to ISO/TS 15066 Regulations. *IEEE Robotics and Automation Letters* 5(4): 5921–5928. doi: 10.1109/LRA.2020.3010494.

Ferraguti F, Preda N, Manurung A, Bonfe M, Lambercy O, Gassert R, Muradore R, Fiorini P and Secchi C (2015) An Energy Tank-Based Interactive Control Architecture for Autonomous and Teleoperated Robotic Surgery. *IEEE Transactions on Robotics* 31(5): 1073–1088.

Ferraguti F, Secchi C and Fantuzzi C (2013) A tank-based approach to impedance control with variable stiffness. *2013 IEEE International Conference on Robotics and Automation* : 4948–4953.

Flacco F and De Luca A (2017) Real-Time Computation of Distance to Dynamic Obstacles With Multiple Depth Sensors. *IEEE Robotics and Automation Letters* 2(1): 56–63. doi: 10.1109/LRA.2016.2535859.

Flacco F, De Luca A and Khatib O (2012) Prioritized multi-task motion control of redundant robots under hard joint constraints. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, Portugal: IEEE, pp. 3970–3977. doi: 10.1109/IROS.2012.6385619.

Flacco F, Kroeger T, De Luca A and Khatib O (2015) A Depth Space Approach for Evaluating Distance to Objects. *Journal of Intelligent & Robotic Systems* 80(S1): 7–22. doi: 10.1007/s10846-014-0146-2.

Folkertsma GA and Stramigioli S (2015) Energy in Robotics. *Foundations and Trends in Robotics* 6(3): 140–210. doi: 10.1561/2300000038.

Frankel T (2011) *The Geometry of Physics*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9781139061377.

Frankel T (2012) *The geometry of physics: an introduction*. 3rd ed edition. Cambridge ; New York: Cambridge University Press. ISBN 978-1-107-60260-1.

Franken M, Stramigioli S, Misra S, Secchi C and Macchelli A (2011) Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency. *IEEE Transactions on Robotics* 27(4): 741–756. doi: 10.1109/TRO.2011.2142430.

Fritzsche M, Elkmann N and Schulenburg E (2011) Tactile sensing: A Key Technology for Safe Physical Human Robot Interaction. *Proceedings of the 6th International Conference on Human Robot Interaction, HRI 2011* : 139.

Garofalo G and Ott C (2018) Passive Energy-based Control via Energy Tanks and Release Valve for Limit Cycle and Compliance Control. *IFAC-PapersOnLine* 51(22): 73–78. doi: 10.1016/j.ifacol.2018.11.520.

Groothuis SS, Folkertsma GA and Stramigioli S (2018) A General Approach to Achieving Stability and Safe Behavior in Distributed Robotic Architectures. *Frontiers in Robotics and AI* 5: 1. doi: 10.3389/frobt.2018.00108.

Haddadin S, Albu-Schaffer A, De Luca A and Hirzinger G (2008a) Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* : 3356–3363.

Haddadin S, Albu-Schaffer A, Frommberger M and Hirzinger G (2008b) The role of the robot mass and velocity in physical human-robot interaction - Part II: Constrained blunt impacts. *2008 IEEE International Conference on Robotics and Automation* : 1339–1345.

Haddadin S, Albu-Schaffer A, Haddadin F, Rosmann J and Hirzinger G (2011) Study on Soft-Tissue Injury in Robotics. *IEEE Robotics & Automation Magazine* 18(4): 20–34. doi: 10.1109/MRA.2011.942996.

Haddadin S, Albu-Schaffer A and Hirzinger G (2008c) The role of the robot mass and velocity in physical human-robot interaction - Part I: Non-constrained blunt impacts. *2008 IEEE International Conference on Robotics and Automation* : 1331–1338.

Haddadin S, Albu-Schäffer A and Hirzinger G (2009) Requirements for Safe Robots: Measurements, Analysis and New Insights. *The International Journal of Robotics Research* 28(11-12): 1507–1527. doi: 10.1177/0278364909343970.

Haddadin S, Albu-Schäffer A and Hirzinger G (eds.) (2010) *Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights*, volume 66. Berlin: Springer. doi: 10.1007/978-3-642-14743-2.

Haddadin S, De Luca A and Albu-Schaffer A (2017) Robot Collisions: A Survey on Detection, Isolation, and Identification. *IEEE Transactions on Robotics* 33(6): 1292–1312. doi: 10.1109/TRO.2017.2723903.

Han H and Park J (2013) Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm. *International Journal of Advanced Robotic Systems* 10(10): 346. doi: 10.5772/56714.

Hermus J, Lachner J, Verdi D and Hogan N (2022) Exploiting redundancy to facilitate physical interaction. *IEEE Transactions on Robotics* 38(1): 599–615. doi: 10.1109/TRO.2021.3086632.

Hjorth S, Lachner J, Stramigioli S, Madsen O and Chrysostomou D (2021) An energy-based approach for the integration of collaborative redundant robots in restricted work environments. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020*. pp. 7152–7158. doi: 10.1109/IROS45743.2020.9341561.

Hogan N (1984) Impedance control of industrial robots. *Robotics and Computer-Integrated Manufacturing* 1(1): 97–113. doi: 10.1016/0736-5845(84)90084-X.

Hogan N (1985a) Impedance Control: An Approach to Manipulation: Part III—Applications. *Journal of Dynamic Systems, Measurement, and Control* 107(1): 17–24. doi: 10.1115/1.3140701.

Hogan N (1985b) Impedance Control: An Approach to Manipulation: Part II—Implementation. *Journal of Dynamic Systems, Measurement, and Control* 107(1): 8–16. doi: 10.1115/1.3140713.

Hogan N (1985c) The mechanics of multi-joint posture and movement control. *Biological Cybernetics* 52(5): 315–331. doi: 10.1007/BF00355754.

Hogan N (1988) On the stability of manipulators performing contact tasks. *IEEE Journal on Robotics and Automation* 4(6): 677–686. doi: 10.1109/56.9305.

Hogan N (2014) A General Actuator Model Based on Nonlinear Equivalent Networks. *IEEE/ASME Transactions on Mechatronics* 19(6): 1929–1939. doi: 10.1109/TMECH.2013.2294096.

Huang HY, Arami A, Farkhatdinov I, Formica D and Burdet E (2020) The Influence of Posture, Applied Force and Perturbation Direction on Hip Joint Viscoelasticity. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28(5): 1138–1145. doi: 10.1109/TNSRE.2020.2983515.

Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P and Schaal S (2013) Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation* 25(2): 328–373. doi: 10.1162/NECO_a_00393.

International Federation of Robotics (2019a) Executive Summary World Robotics 2019: Industrial Robots. Technical report, International Federation of Robotics, Frankfurt am Main. URL `https://ifr.org/free-downloads`.

International Federation of Robotics (2019b) Executive Summary World Robotics 2019: Service Robots. Technical report, International Federation of Robotics, Frankfurt am Main. URL `https://ifr.org/free-downloads`.

International Organization for Standardization (2010) ISO 12100: Safety of machinery: General principles for design: Risk assessment and risk reduction. Technical report, International Organization for Standardization.

International Organization for Standardization (2011) ISO 10218-1: Robots and robotic devices: Safety requirements for industrial robots - Part 1: Robots. Technical report, International Organization for Standardization.

International Organization for Standardization (2015) ISO 13849-1: Safety of machinery: Safety-related parts of control systems; Part 1: General principles for design. Technical report, International Organization for Standardization.

International Organization for Standardization (2016) ISO-TS 15066: Robots and robotic devices: Collaborative robots. Technical report, International Organization for Standardization.

Kanoun O, Lamiraux F and Wieber PB (2011) Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task. *IEEE Transactions on Robotics* 27(4): 785–792. doi: 10.1109/TRO.2011.2142450.

Khalil W and Dombre E (2002) In: *Modeling, Identification and Control of Robots*. Oxford: Butterworth-Heinemann. doi: https://doi.org/10.1016/B978-190399666-9/50005-1.

Khan SG, Herrmann G, Al Grafi M, Pipe T and Melhuish C (2014) Compliance Control and Human–Robot Interaction: Part 1 — Survey. *International Journal of Humanoid Robotics* 11(03): 1430001. doi: 10.1142/S0219843614300013.

Khatib O (1980) *Commande dynamique dans l'espace operationnel des robots manipulateurs en presence d'obstacles.* PhD Thesis, Universite Paul Sabatier, Toulouse.

Khatib O (1987) A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation* 3(1): 43–53. doi: 10.1109/JRA.1987.1087068.

Khatib O (1995) Inertial Properties in Robotic Manipulation: An Object-Level Framework. *The International Journal of Robotics Research* 14(1): 19–36. doi: 10.1177/027836499501400103.

Khatib O and Sentis L (2004) Whole-Body Dynamic Behavior and Control of Human-like Robots. *International Journal of Humanoid Robotics* 01(01): 29–43. doi: 10.1142/S0219843604000058.

Klein CA and Blaho BE (1987) Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators. *The international journal of robotics research* 6(2): 72–83. doi: 10.1177/027836498700600206.

Lachner J, Allmendinger F, Hobert E, Hogan N and Stramigioli S (2021) Energy budgets for coordinate invariant robot control in physical human–robot interaction. *The International Journal of Robotics Research* 40(8-9): 968–985. doi: 10.1177/02783649211011639.

Lachner J, Allmendinger F, Stramigioli S and Hogan N (2022) Shaping impedances to comply with constrained task dynamics. *IEEE Transactions on Robotics* doi: 10.1109/TRO.2022.3153949.

Lachner J, Schettino V, Allmendinger F, Fiore MD, Ficuciello F, Siciliano B and Stramigioli S (2020) The influence of coordinates in robotic manipulability analysis. *Mechanism and Machine Theory* 146. doi: 10.1016/j.mechmachtheory.2019.103722.

Lachner J and Stramigioli S (2021) Method and system for operating a robot. URL `https://patents.google.com/patent/DE102020209866B3`.

Lacquaniti F, Carrozzo M and Borghese NA (1993) Time-varying mechanical behavior of multijointed arm in man. *Journal of Neurophysiology* 69(5): 1443–1464. doi: 10.1152/jn.1993.69.5.1443.

Laffranchi M, Tsagarakis NG and Caldwell DG (2009) Safe human robot interaction via energy regulation control. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* : 35–41.

Lang S (1987) *Linear Algebra*. Undergraduate Texts in Mathematics. New York, NY: Springer New York. doi: 10.1007/978-1-4757-1949-9.

Laskey M, Staszak S, Hsieh WYS, Mahler J, Pokorny FT, Dragan AD and Goldberg K (2016) SHIV: Reducing supervisor burden in DAgger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, pp. 462–469. doi: 10.1109/ICRA.2016.7487167.

Lee H, Rouse EJ and Krebs HI (2016) Summary of Human Ankle Mechanical Impedance During Walking. *IEEE Journal of Translational Engineering in Health and Medicine* 4: 1–7. doi: 10.1109/JTEHM.2016.2601613.

Lee SD and Song JB (2015) Collision detection of humanoid robot arm under model uncertainties for handling of unknown object. *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* : 718–721.

Levi E (1905) Sulla struttura dei gruppi finiti e continui. *Atti della Reale Accademia delle scienze di Torino* 40: 551–565.

Liang J, Wu J, Huang H, Xu W, Li B and Xi F (2020) Soft Sensitive Skin for Safety Control of a Nursing Robot Using Proximity and Tactile Sensors. *IEEE Sensors Journal* 20(7): 3822–3830. doi: 10.1109/JSEN.2019.2959311.

Lipkin H and Duffy J (1988) Hybrid Twist And Wrench Control for a Robotic Manipulator. *Trans. ASME J. Mech. Transm. Automation Design* 110: 138–144.

Ma O and Angeles J (1991) Optimum architecture design of platform manipulators. *International Conference on Advanced Robotics 'Robots in Unstructured Environments* 5: 1130–1135 vol.2. doi: 10.1109/ICAR.1991.240404.

Manschitz S, Gienger M, Kober J and Peters J (2020) Learning Sequential Force Interaction Skills. *Robotics* 9(2): 45. doi: 10.3390/robotics9020045.

Martin-Martin R, Lee MA, Gardner R, Savarese S, Bohg J and Garg A (2019) Variable Impedance Control in End-Effector Space: An Action Space for Reinforcement Learning in Contact-Rich Tasks. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, pp. 1010–1017. doi: 10.1109/IROS40897.2019.8968201.

Maurice P, Hogan N and Sternad D (2018) Predictability, force, and (anti)resonance in complex object control. *Journal of Neurophysiology* 120(2): 765–780. doi: 10.1152/jn.00918.2017.

McIntyre J, Mussa-Ivaldi F and Bizzi E (1996) The control of stable postures in the multijoint arm. *Experimental Brain Research* 110(2). doi: 10.1007/BF00228556.

Merlet JP (2006) Jacobian, Manipulability, Condition Number and Accuracy of Parallel Robots. *Journal of Mechanical Design 128* 1: 199–206. doi: 10.1007/978-3-540-48113-316.

Murray RM, Li Z and Sastry S (1994) *A mathematical introduction to robotic manipulation.* Boca Raton: CRC Press. ISBN 978-0-8493-7981-9.

Mussa-Ivaldi FA and Hogan N (1991) Integrable Solutions of Kinematic Redundancy via Impedance Control. *The International Journal of Robotics Research* 10(5): 481–491. doi: 10.1177/027836499101000504.

Muñoz Osorio JD, Castañeda F, Allmendinger F and Zimmermann UE (2019) Time Invariant Motion Controller for Physical Human Robot Interaction. *43rd Mechanisms and Robotics Conference* doi: 10.1115/DETC2019-98031.

Nakamura Y, Hanafusa H and Yoshikawa T (1987) Task-Priority Based Redundancy Control of Robot Manipulators. *The international journal of robotics research* 6(2): 3–15. doi: 10.1177/027836498700600201.

Natale C (ed.) (2003) *Interaction Control of Robot Manipulators: Six degrees-of-freedom tasks*, *Springer Tracts in Advanced Robotics*, volume 3. Berlin and Heidelberg: Springer. doi: 10.1007/3-540-36155-3.

Navarro B, Cherubini A, Fonte A, Passama R, Poisson G and Fraisse P (2016) An ISO10218-compliant adaptive damping controller for safe physical human-robot interaction. *2016 IEEE International Conference on Robotics and Automation (ICRA)* : 3043–3048.

Osorio JDM, Allmendinger F, Fiore MD, Zimmermann UE and Ortmaier T (2019) Physical Human-Robot Interaction under Joint and Cartesian Constraints. In: *2019 19th International Conference on Advanced Robotics (ICAR).* Belo Horizonte, Brazil, pp. 185–191. doi: 10.1109/ICAR46387.2019.8981579.

Ott C (2008) *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, volume 49. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-69255-3.

Paden B and Sastry S (1988) Optimal Kinematic Design of 6R Manipulators. *The International Journal of Robotics Research* 7(2): 43–61. doi: 10.1177/027836498800700204.

Park FC (1995) Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design. *Journal of Mechanical Design* 117(1): 48–54. doi: 10.1115/1.2826116.

Park FC and Brockett RW (1994) Kinematic Dexterity of Robotic Mechanisms. *The International Journal of Robotics Research* 13(1): 1–15. doi: 10.1177/027836499401300101.

Park FC, Murray AP and McCarthy JM (1993) Designing Mechanisms for Workspace Fit. In: Angeles J, Hommel G and Kovács P (eds.) *Computational Kinematics*. Dordrecht: Springer Netherlands, pp. 295–306. doi: 10.1007/978-94-015-8192-9_27.

Paul RP and Stevenson CN (1983) Kinematics of Robot Wrists. *The international journal of robotics research* 2(1): 31–38. doi: 10.1177/027836498300200103.

Paynter HM, Briggs P and Massachusetts Institute of Technology (1961) *Analysis and design of engineering systems: class notes for M.I.T. course 2.751*. Cambridge, Mass.: M.I.T. Press. URL `//catalog.hathitrust.org/Record/005135763`.

Penrose R (1955) A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society* 51(3): 406–413. doi: 10.1017/S0305004100030401.

Perreault E, Kirsch R and Crago P (2004) Multijoint dynamics and postural stability of the human arm. *Experimental Brain Research* 157(4). doi: 10.1007/s00221-004-1864-7.

Phan S, Quek ZF, Shah P, Shin D, Ahmed Z, Khatib O and Cutkosky M (2011) Capacitive skin sensors for robot impact monitoring. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* : 2992–2997.

Raiola G, Cardenas CA, Tadele TS, de Vries T and Stramigioli S (2018) Development of a Safety- and Energy-Aware Impedance Controller for Collaborative Robots. *IEEE Robotics and Automation Letters* 3(2): 1237–1244. doi: 10.1109/LRA.2018.2795639.

Rancourt D and Hogan N (2001) Stability in Force-Production Tasks. *Journal of Motor Behavior* 33(2): 193–204. doi: 10.1080/00222890109603150.

Rodrigues O (1816) *De l'attraction des sphéroides*. PhD Thesis, Université de Paris, Paris. doi: 10.3990/1.9789036537841.

Rosenstrauch MJ and Kruger J (2017) Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066. *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)* : 740–744.

Sadrfaridpour B and Wang Y (2018) Collaborative Assembly in Hybrid Manufacturing Cells: An Integrated Framework for Human–Robot Interaction. *IEEE Transactions on Automation Science and Engineering* 15(3): 1178–1192. doi: 10.1109/TASE.2017.2748386.

Salisbury JK and Craig JJ (1982) Articulated Hands: Force Control and Kinematic Issues. *The international journal of robotics research* 1(1): 4–17. doi: 10.1177/027836498200100102.

Schaal S (2006) Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics. In: Kimura H, Tsuchiya K, Ishiguro A and Witte H (eds.) *Adaptive Motion of Animals and Machines*. Tokyo: Springer Tokyo, pp. 261–280. doi: 10.1007/4-431-31381-8_23.

Schettino V, Fiore M, Pecorella C, Ficuciello F, Allmendinger F, Lachner J, Stramigioli S and Siciliano B (2021) Geometrical interpretation and detection of multiple task conflicts using a coordinate invariant index. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020*. pp. 6613–6618. doi: 10.1109/IROS45743.2020.9340690.

Scheurer C, Fiore MD, Sharma S and Natale C (2016) Industrial implementation of a multi-task redundancy resolution at velocity level for highly redundant mobile manipulators. *47th International Symposium on Robotics (ISR 2016)* 1(1).

Schindlbeck C and Haddadin S (2015) Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. *2015 IEEE International Conference on Robotics and Automation (ICRA)* : 440–447.

Schreiber G, Stemmer A and Bischoff R (2010) The Fast Research Interface for the KUKA Lightweight Robot. *IEEE ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications* 1(1).

Schwartz EM, Manseur R and Doty KL (2002) Noncommensurate Systems in Robotics. *International Journal of Robotics and Automation* 17(2).

Sentis L and Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics* 02(04): 505–518. doi: 10.1142/S0219843605000594.

Sentis L and Khatib O (2006) A whole-body control framework for humanoids operating in human environments: Operating in Human Environments. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* doi: 10.1109/ROBOT.2006.1642100.

Shahriari E, Johannsmeier L and Haddadin S (2018) Valve-based virtual energy tanks: A framework to simultaneously passify controls and embed control objectives. In: *2018 Annual American Control Conference (ACC)*. pp. 3634–3641. doi: 10.23919/ACC.2018.8431718.

Shahriari E, Johannsmeier L, Jensen E and Haddadin S (2020) Power Flow Regulation, Adaptation, and Learning for Intrinsically Robust Virtual Energy Tanks. *IEEE Robotics and Automation Letters* 5(1): 211–218. doi: 10.1109/LRA.2019.2953662.

Shahriari E, Kramberger A, Gams A, Ude A and Haddadin S (2017) Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives. *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)* : 136–142.

Shuster MD (1993) A Survey of Attitude Representation. *The Journal of the Astronautical Sciences* 41(4): 439–517.

Siciliano B and Khatib O (eds.) (2008) *Springer Handbook of Robotics.* Berlin, Heidelberg: Springer Science+Business Media. doi: 10.1007/978-3-540-30301-5.

Siciliano B, Sciavicco L, Villani L and Oriolo G (2009) *Robotics: Modelling, Planning and Control.* London: Springer London. doi: 10.1007/978-1-84628-642-1.

Siciliano B and Slotine JJ (1991) A general framework for managing multiple tasks in highly redundant robotic systems. In: *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments.* Pisa, Italy: IEEE, pp. 1211–1216 vol.2. doi: 10.1109/ICAR.1991.240390.

Stramigioli S (2001) *Modeling and IPC control of interactive mechanical systems - A coordinate-free approach*, *Lecture Notes in Control and Information Sciences*, volume 266. London: Springer. doi: 10.1007/BFb0110400.

Stramigioli S (ed.) (2015) *Energy-Aware Robotics*, *Mathematical Control Theory I*, volume 461. Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-20988-3.

Stramigioli S and Bruyninckx H (2001a) Geometry and Screw Theory for Robotics. *Robotics and Autonomous Systems* 36(193): 194.

Stramigioli S and Bruyninckx H (2001b) Geometry of dynamic and higher-order kinematic screws. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, volume 4. Seoul, South Korea: IEEE, pp. 3344–3349. doi: 10.1109/ROBOT.2001.933134.

Su H, Li S, Manivannan J, Bascetta L, Ferrigno G and Momi ED (2019) Manipulability Optimization Control of a Serial Redundant Robot for Robot-assisted Minimally Invasive Surgery. *International Conference on Robotics and Automation* 1: 1323–1328. doi: 10.1109/ICRA.2019.8793676.

Tadele TS, de Vries TJA and Stramigioli S (2014) Combining energy and power based safety metrics in controller design for domestic robots. *2014 IEEE International Conference on Robotics and Automation (ICRA)* : 1209–1214.

Takegaki M and Arimoto S (1981) A New Feedback Method for Dynamic Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control* 103(2): 119–125. doi: 10.1115/1.3139651.

Tortora GJ and Derrickson BH (2017) *Principles of Anatomy and Physiology.* John Wiley. ISBN 978-1-119-40006-6.

van der Schaft A (2016) *L2-Gain and Passivity Techniques in Nonlinear Control*. 3rd edition. Springer Publishing Company, Incorporated. ISBN 3319499912.

Verdi D (2017) *A Compositional Approach to Robotic Impedance Control*. Master's Thesis, Massachusetts Institute of Technology.

Wassink M and Stramigioli S (2007) Towards a novel safety norm for domestic robotics. *IEEE/RSJ International Conference on Intelligent Robots and Systems* doi: 10.1109/IROS.2007.4399199.

Willems J (1972) Dissipative dynamical systems part i: General theory. *Archive for Rational Mechanics and Analysis* 45(5): 1432–0673. doi: 10.1007/BF00276493.

Wu X (2016) *Passivation of a Hierarchical Whole-Body Controller for Humanoid Robots*. Master's Thesis, Technische Universität München.

Wyatt J, Chua L, Gannett J, Goknar I and Green D (1981) Energy concepts in the state-space theory of nonlinear n-ports: Part I-Passivity. *IEEE Transactions on Circuits and Systems* 28(1): 48–61. doi: 10.1109/TCS.1981.1084907.

Yoshikawa T (1985a) Dynamic manipulability of robot manipulators. *Proceedings of the 1985 International Conference on Robotics and Automation* 1: 1033–1038. doi: 10.1109/ROBOT.1985.1087277.

Yoshikawa T (1985b) Manipulability of Robotic Mechanisms. *The international journal of robotics research* 4(2): 3–9. doi: 10.1177/027836498500400201.

Zefran M and Kumar V (1997) Affine connections for the cartesian stiffness matrix. In: *Proceedings of International Conference on Robotics and Automation*, volume 2. pp. 1376–1381. doi: 10.1109/ROBOT.1997.614329.

Zefran M, Kumar V and Croke C (1999) Metrics and Connections for Rigid-Body Kinematics. *The International Journal of Robotics Research* 18(2): 242–1. doi: 10.1177/027836499901800208.