Lehrstuhl für Kognitive Systeme
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Set Point Generation and Task Specification for Indirect Force Controlled Robots

Dipl.-Ing. Univ. Ewald Lutscher

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr.-Ing. habil. Dirk Wollherr
Prüfer der Dissertation:	1.	Prof. Dr. Gordon Cheng, Ph.D.
	2.	Prof. Dr. Alin Albu-Schäffer

Die Dissertation wurde am 20.06.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 23.03.2017 angenommen.

Abstract

Future robots are expected to work in a highly unstructured environment (e.g. domestic robots). Compliance is a compulsory requirement for such robots and a common approach to realize this is *indirect force control* (IFC), e.g. impedance control. In most applications, conventional trajectory planning strategies are applied for indirect force controlled robots with the IFC having the sole purpose of compensating for the uncertain task geometry or playing the role of a safety buffer for unexpected collisions. The main goal of this thesis is to investigate different approaches to set-point generation for indirect force controlled robots, which go beyond these conventional use cases and aim for a more direct task oriented approach.

During the work for this thesis, a new control layer for indirect force controlled robots was developed, which provides application programmers with an intuitive interface to implement complex tasks, involving positioning and force components on joint and Cartesian level. The resulting task specification framework allows to specify a set of hierarchically ordered equality or inequality subtasks to program a desired behavior of the robot while obeying its physical limits. The functionality and performance is validated experimentally.

In addition, the problem of manipulating unknown and highly kinematically constrained mechanisms, like doors or drawers, is approached using IFC. A major requirement was high generality in terms of operated mechanisms and robotic hardware. Based on a simple strategy for an admittance controlled mobile manipulator, it is first shown how the method can be further generalized to a wider class of manipulators by considering joint-level IFC's in addition. Then, the task specification framework, developed in the previous chapters, is applied to further improve the performance, regarding erroneous forces, by proper separation of the force and positioning subtasks.

The main contributions of this thesis can be summarized as follows:

1. Investigation of task oriented set-point generation methods for indirect force con-

trolled robots.

2. Development of an IFC task specification formalism, involving equality and inequality force and positioning components on joint and Cartesian level.
3. Implementation of various use cases on different robotic platforms to highlight the generality of the developed method and demonstrate the practical relevance.
4. Presentation of a strategy for manipulation of unknown constrained mechanisms using indirect force control.

Kurzfassung

Von zukünftigen Robotern wird erwartet, dass sie in einer hochgradig unstrukturierten Umgebung arbeiten (z.B. Haushaltsroboter). Nachgiebigkeit ist eine obligatorische Anforderung für solche Roboter und ein üblicher Ansatz diese zu realisieren, ist indirekte Kraftregelung (IFC), z.B. Impedanzregelung. Die meisten Applikationen verwenden konventionelle Trajektorienplanungsstrategien für indirekt kraftgeregelte Roboter und der IFC dient nur dem Zweck, unsichere Taskgeometrie zu kompensieren oder er spielt die Rolle eines Sicherheitspuffers für unerwartete Kollisionen. Das Hauptziel dieser Dissertation ist es, verschiedene Ansätze für die Sollwertgenerierung für indirekt kraftgeregelte Roboter zu untersuchen, die über diese konventionellen Anwendungsfälle hinausgehen und auf einen Task-orientierteren Ansatz abzielen.

Während der Arbeit für diese Dissertation wurde eine neue Steuerungsebene für indirekt kraftgeregelte Roboter entwickelt, die Anwendungsprogrammierern eine intuitive Schnittstelle zur Verfügung stellt um komplexe Tasks zu implementieren, die Positionierungs- als auch Kraftkomponenten auf Gelenk- und kartesischer Ebene enthalten. Das resultierende Task-Spezifizierungs-Framework erlaubt es, einen Satz hierarchisch geordneter Gleichungs- und Ungleichungs-Subtasks zu spezifizieren um ein gewünschtes Verhalten des Roboters unter Einhaltung seiner physikalischen Begrenzungen zu programmieren. Die Funktionalität und Performanz wurde experimentell validiert.

Zusätzlich wird das Problem der Manipulation von unbekannten und hochgradig restringierten Mechanismen, wie Türen und Schubladen, unter Verwendung von IFC's, angegangen. Eine wichtige Anforderung war hohe Generalisierbarkeit bezüglich der Mechanismen und der Roboter-Hardware. Basierend auf einer einfachen Strategie für einen admittanzgeregelten mobilen Manipulator, wird zuerst gezeigt, wie die Methode auf eine breitere Klasse von Manipulatoren weiter vereinheitlicht werden kann durch zusätzliche Berücksichtigung von IFC's auf Gelenkebene. Im Anschluss wird durch geeignete Trennung in Kraft- und Positionierungs-Subtasks, das in den vorigen Kapiteln entwickelte, Task-Spezifizierungs-Framework genutzt um die Performanz im Hinblick auf

Fehlkräfte, weiter zu verbessern.

Die Hauptbeiträge dieser Arbeit können folgendermaßen zusammengefasst werden:

1. Untersuchung von Taskorientierten Sollwert-Generierungsmethoden für indirekt Kraftgeregelte Roboter

2. Entwicklung eines IFC Task-Spezifizierungsformalismus, der Gleichungs- und Ungleichungs-Kraft- und Positionierungskomponenten auf Gelenk- und auf Kartesischer Ebene beinhaltet.

3. Implementierung von mehreren Anwendungsfällen auf verschiedenen Roboterplattformen um die Allgemeingültigkeit der entwickelten Methode hervorzuheben und ihre praktische Relevanz zu demonstrieren

4. Vorstellung einer Strategie für die Manipulation von unbekannten, kinematisch restringierten Mechanismen unter Verwendung von indirekter Kraftregelung.

Acknowledgements

First of all, I would like to thank my supervisor Gordon Cheng, who gave me the opportunity to carry out my research as a PhD student. He always encouraged me and gave me great autonomy while being constantly supportive. I am also very grateful to all of my colleagues at ICS for their input and help, especially Andreas Holzbach, who shared the office and his wisdom with me.

I gratefully acknowledge the funding from the DFG cluster of excellence Cognition for Technical Systems (CoTeSys), which made my PhD possible.

My special thanks go to my parents Ewald and Nathalia for their great support in all live situations.

Munich, April 29, 2017

Contents

Notations

Abbreviations

min.	minimize
s.t.	subject to
DoF	Degree of Freedom
KKT	Karush-Kuhn-Tucker conditions
FK	Forward Kinematics
IK	Inverse Kinematics
inst. IK	instantaneous Inverse Kinematics
SMA	Simple Moving Average
IFC	Indirect Force Control
SPG	Set-Point Generator
PSPM	Passive Set-Point Modulation
RMS	Root Mean Square
RCC	Remote Center of Compliance
FTS	Force Torque Sensor

Conventions

Scalars are italicized in both upper and lower cases. Vectors are denoted by lower case letters in italicized boldface style. Matrices are denoted by upper case letters in italicized boldface style.

x, X	scalar
$\boldsymbol{x}$	vector
$\boldsymbol{X}$	matrix
$\dot{x}, \ddot{x}$	equivalent to $\frac{d}{dt}x, \frac{d^2}{dt^2}x$
$\hat{x}$	estimated value of x
$\tilde{x}$	deviation from desired value $x_d - x$
$\boldsymbol{X}^{-1}$	inverse
$\boldsymbol{X}^{\#}$	generalized inverse
$\boldsymbol{X}^{+}$	Moore-Penrose pseudo-inverse
$\boldsymbol{X}^{T}$	Transposed of $\boldsymbol{X}$
$\boldsymbol{I}_r/\mathbf{0}_r$	$r \times r$ identity/zero matrix
$\mathbf{0}_{r\times s}$	$r \times s$ zero matrix
$\mathbf{1}_{r\times s}$	$r \times s$ matrix of ones

Sub- and Superscripts

x_d	desired value
x_m/x_M	lower/upper limit
x^*	optimal solution of an optimization problem
$\boldsymbol{x}_{ee}$	Cartesian pose of the end-effector

Symbols

i/j	general iteration variables
n	number of joints
m	task dimension
k	number of tasks
$\boldsymbol{x}$	Cartesian pose
$\boldsymbol{q}$	vector of joint angles

$\boldsymbol{K}$	virtual joint stiffness matrix
$\boldsymbol{D}$	virtual joint damping matrix
$\boldsymbol{K}_x$	virtual Cartesian stiffness matrix
$\boldsymbol{K}_p$	translational components of virtual Cartesian stiffness matrix
$\boldsymbol{K}_o$	rotational components of virtual Cartesian stiffness matrix
$\boldsymbol{M}$	joint space mass matrix
$\boldsymbol{c}$	vector of torques due to Coriolis and centrifugal forces
$\boldsymbol{\tau}_g$	vector of gravitational torques
$\boldsymbol{\omega}$	angular velocity
$\boldsymbol{p}$	position vector
$\boldsymbol{o}$	general vector of orientation representation
$\boldsymbol{\tau}$	vector of commanded joint torques
$\boldsymbol{\tau}_s$	static components of $\boldsymbol{\tau}$
$\boldsymbol{\tau}_{\text{ext}}$	external torques
$\boldsymbol{f}$	force vector
$\boldsymbol{m}$	moment vector
$\boldsymbol{h}$	wrench due to commanded joint torques
$\boldsymbol{h}_s$	static components of $\boldsymbol{h}$
$\boldsymbol{q}_v$	vector of virtual joint angles
$\boldsymbol{J}$	manipulator's base Jacobian
$\boldsymbol{J}_v$	virtual manipulator's base Jacobian
$\boldsymbol{J}_c$	general contact Jacobian
$\boldsymbol{S}$	task selection matrix
$\boldsymbol{A}$	general task Jacobian
μ	manipulability index
$\boldsymbol{\alpha}$	general task variable
$\gamma(\dot{\boldsymbol{q}})$	differential compensation for joint velocities
$\dot{\boldsymbol{\beta}}$	compensated differential task variable $\dot{\boldsymbol{\alpha}} + \gamma(\dot{\boldsymbol{q}})$
$\boldsymbol{N}(\boldsymbol{X})$	nullspace-mapping operator
E	system energy
E_K	energy stored in the virtual spring
E_D	energy dissipated by the virtual damper
E_r	energy stored in the virtual energy reservoir
ΔT	time interval of the discrete controller

Functions

$\mathrm{FK}(\boldsymbol{q})$	Forward Kinematics
$\mathrm{rank}(\boldsymbol{X})$	rank of $\boldsymbol{X}$
$\mathrm{norm}(\boldsymbol{x})$	normalization $\frac{\boldsymbol{x}}{\lvert\lvert\boldsymbol{x}\rvert\rvert}$
$\ker(\boldsymbol{X})$	kernel/nullspace of a linear map $\boldsymbol{X}$
$\mathrm{diag}(\boldsymbol{X}, \boldsymbol{Y})$	block diagonal matrix composed of the two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$
$\det(\boldsymbol{X})$	determinant of $\boldsymbol{X}$
$\min/\max(a, b, c, \dots)$	returns the minimum/maximum of a set of input values

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1. Motivation and Problem Description

When operating in unstructured environments, compliance is an important requirement for a robotic manipulator. A stable and robust approach to realize compliance is provided by indirect force controllers, where the motion and interaction forces of the physical robot are indirectly controlled by regulating the motion of a virtual robot, which on his part is coupled with the physical robot via a virtual mechanical relationship. This control scheme has also nice stability properties which are mostly independent from the environmental dynamics, what makes it perfectly suitable for physical interaction with an uncertain environment. Due to these advantages, IFC's are provided often as the only force control interface for robots (closed architecture). Fig. 1.1 depicts the basic idea behind IFC.

The set-point selection for the virtual robot is also referred to as virtual trajectory generation and while there is an extensive amount of work, having the purpose of improving the rendering of the virtual mechanical relationship or proposing different extensions or variations for IFC's (for impedance control in particular), the literature covering virtual trajectory generation is very sparse. Traditional trajectory planning approaches are often applied and the IFC is used to compensate for contact uncertainty and unexpected collisions without explicitly considering the applied interaction forces (see Fig. 1.2). Some IFC related works focus on accurate force tracking [104, 49] and present a narrowed application range which could also be achieved by direct force control. In addition, these and similar works focus mostly on Cartesian-level IFC's.

The major goal of this thesis is to develop a task specification scheme for joint-level indirect force controlled robots for tasks consisting of force and positioning components on joint- and Cartesian-level. By providing a sound solution for joint-level IFC's, the

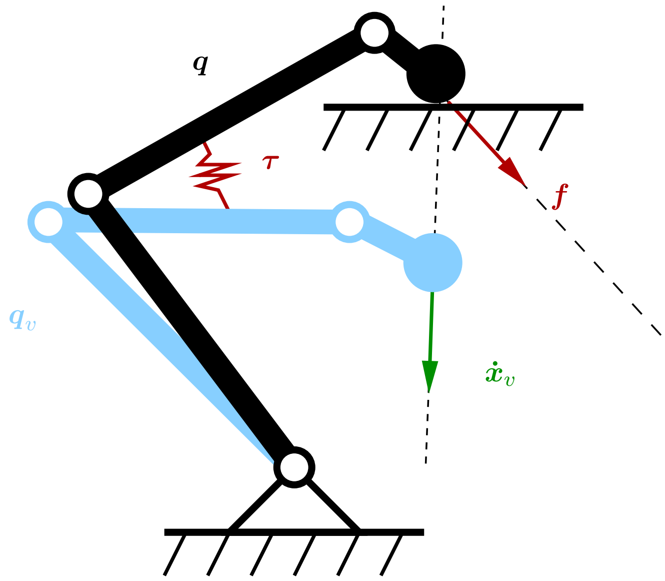


Figure 1.1. Motion and interaction forces of the physical manipulator (black) are controlled indirectly by generating set-points for the virtual manipulator (blue) according to a virtual mechanical relationship.

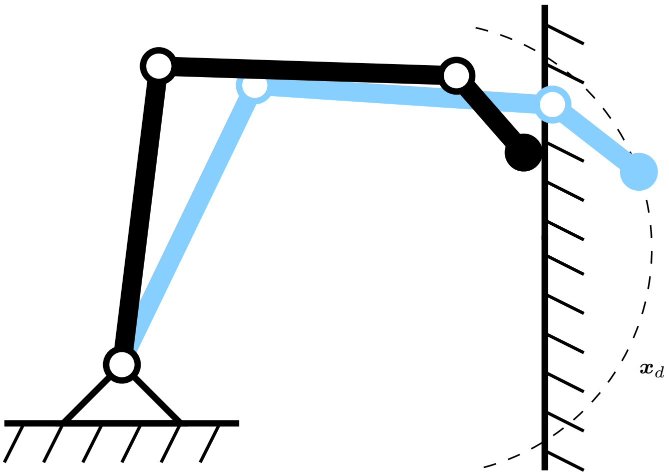


Figure 1.2. Conventional application of IFC's. The inherent compliance is used to compensate for unexpected interaction forces while tracking a desired trajectory $\boldsymbol{x}_d$.

developed scheme is in general also applicable on *serial-elastic actuator* (SEA) robots. In fact, joint space IFC is the only way of regulating the interaction forces for a serial-elastic actuator robot with position controlled joints. The developed scheme should also fulfill the following requirements:

- The fundamental approach should be in general independent from the robotic hardware with the IFC as abstraction layer and go without additional external sensing.
- It should be easy to program real world tasks without detailed knowledge of the underlying low-level control structure.

1.2. Related Work

1.2.1. Simultaneous Force and Position Control

Due to the limited capabilities of sensors and motors, the first robotic applications were purely motion controlled. Even nowadays, where the research on robot force control has flourished for decades, industrial robots still largely rely on pure motion control, which is sufficient for most industrial robotic tasks, like welding or painting. To overcome geometric uncertainties for physical interaction tasks, a *remote center of compliance* (RCC) is often used, which is a cheap and simple solution [23]. The drawbacks of this passive approach is that it can only deal with small position and orientation deviations and lacks flexibility, since for every robotic task a custom RCC has to be designed.

Therefor, and also considering tasks where the information on the environment is prone to large uncertainties, respectively where humans are involved, ensuring a compliant behavior of the robot is crucial to avoid damage or injuries. The most appropriate approaches can be put in the category of direct and indirect force control and will be presented in the following.

1.2.1.1. Direct Force Control

Hybrid Force/Position Control The most common approach for simultaneous force and position control is the hybrid force/position control paradigm, originally introduced in [90] and is based on the compliance frame modeling [74]. The underlying assumption is, that it is possible to divide a robotic task into orthogonal force and position controlled subtasks, where the force controlled directions are physically constrained, while the position controlled directions are not. General contact tasks can be realized by specifying the force and position controlled directions. Many implementations of hybrid force/position control exist, e.g. based on operational space control, applying inverse dynamics [57], passivity-based control [113] or outer force loops, closed around inner motion loops [27].

While allowing accurate position and force tracking, a major drawback of this approach is that the task geometry has to be known at least approximately to ensure safe task execution, as the robot in general lacks compliance along the position controlled directions. This issue is especially eminent for a dynamic environment, which is encountered in service robotics, where the task geometry can change abruptly even though it was estimated accurately beforehand, e.g. due to human intervention.

While the problem of controlling force and position simultaneously was already recognized in [32], in highly unstructured environments it is difficult to clearly define pure force or positioning subtasks. An example is opening a spring loaded door, where a usually unknown interaction force has to be applied in order to operate the mechanism, while simultaneously regulating the pose of the end-effector along an uncertain trajectory.

Parallel Control This approach does not explicitly require decoupled force- and position-controlled directions [20, 107]. The force commands are internally transformed into motion commands and weighted by their respective feedback control constants. Force regulation is guaranteed at the expense of position errors by giving more weight to force control.

1.2.1.2. Indirect Force Control

Differently from the direct force control approaches above, IFC does not require an explicit model of the interaction task. Instead of regulating the contact forces directly, a virtual mechanical relationship is imposed between the desired and actual configuration of the robot. Usually this is realized in operational space by using a wrist-mounted force-torque sensor but due to the need for full-body compliance, joint space implementations are getting more and more common, e.g. for the *iCub* humanoid robot [42, 97] or the *Kuka Lightweight Robot* (LBR) [11].

The original concept of generalized spring and damping for force control in joint coordinates was proposed in [82] and the implementation discussed in [115]. The most common IFC variation is impedance control [46]. Impedance control establishes a virtual mass-spring-damper relationship between the desired and actual configuration of the robot, respectively end-effector pose. The stability of impedance control was analyzed in [45]. A similar concept is applied in admittance control [54], where an outer force loop is closed around the inner motion control loop. Special cases of impedance and admittance control consider only the static relationship between the positioning error and the contact forces. These schemes are referred to as stiffness and compliance control respectively [96].

An intuitive interpretation of IFC is that the user specifies the set-points for a virtual manipulator to which the physical manipulator is attached via a virtual mechanical relationship. Therefore, the actual interaction force is a result of how deep the virtual manipulator "penetrates" the environment (1.1).

The major drawback of poor accuracy, is outplayed by the increased interaction safety and robustness to environmental uncertainties and unexpected collisions. Also, IFC schemes do in principle not require measurements of contact forces and moments, which makes them an effective and cheap approach to compliant control for robots lacking force/torque sensing.

Virtual Trajectory Generation The major contributions regarding IFC have the intention to improve the performance of the controller itself or introducing different variants and extensions [22, 66, 2]. Several investigations have also been made targeting the construction of compliances, optimized for specific tasks [85, 99].

Considering applications, conventional trajectory planning approaches are often applied and the IFC is used to compensate for contact uncertainty and unexpected collisions. The only works explicitly dealing with virtual trajectory generation aim at pure force tracking or are part of a customized approach to a specific problem. In [8, 49, 50, 59, 104, 108], force tracking in impedance control for industrial applications is treated. The focus of these works lies on setting proper virtual trajectories to adapt to the unknown environmental stiffness. In [48, 47], where virtual trajectory generation is referred to as equilibrium point control, the virtual trajectory of the manipulator is designed to follow the estimated traversable Cartesian trajectory to open doors and drawers. The compliance is used to compensate for uncertainties in the constraint estimator and the applied interaction forces are not explicitly considered.

1.2.2. Task Specification / Task Programming

Early robots were only position controlled and redundant mechanisms were avoided to reduce the complexity of the system. Hence, the first robotic tasks were simple positioning tasks, trajectory following or pick-and-place applications in well structured environments. Such tasks aim only at positioning and a wide range of specification tools exists and is available in commercial robot control software.

The first concept of assembling a mixed components main task from different subtasks was derived in conjunction with the task frame formalism [74] and further developed in [26, 27, 15]. A force or positioning subtask is assigned to every direction of the specified task frame and a hybrid force/position controller is used to track the desired trajectories or set-points simultaneously. Tasks are denoted as artificial constraints. A survey on such constrained based task specification formalisms can be looked up in [25].

As the hybrid force/position control scheme is used as the underlying low-level control, the task geometry has to be known in advance, respectively estimated online as done in [24, 62].

1.2.3. Contribution of the Present Work

With their robustness to environmental uncertainties and unexpected collisions, IFC's represent a very suitable approach to compliant robot control in highly unstructured and human inhabited environments, like in service robotics. The difficulty of incorporating conventional task specification and planning schemes has detained robotics researchers from using this control scheme, except for tasks where only a certain compliance is required for the sake of safety.

Therefor, IFC has not received much attention in recent years, respectively the only effort was done towards improvement of the control performance or extend the present schemes with additional functionalities.

The present work is meant to fill the gap between the low-level control design and application programming by introducing an additional control layer, which is called *set-point-generator* (SPG). The purpose of the SPG is to provide set-points for the virtual manipulator according to some higher-level task description. Fig. 1.3 depicts how the SPG is connected to a typical IFC loop.

The goal is not to introduce another low-level force controller, instead the underlying IFC is treated as a black box, where only the relation between the configurations of the virtual and the physical robot is considered. Based on these assumptions, it is possible to derive an elaborate set-point generation method to track different mixed force and positioning tasks in joint and Cartesian space simultaneously.

The tasks are defined by the application programmer via a task specification interface,

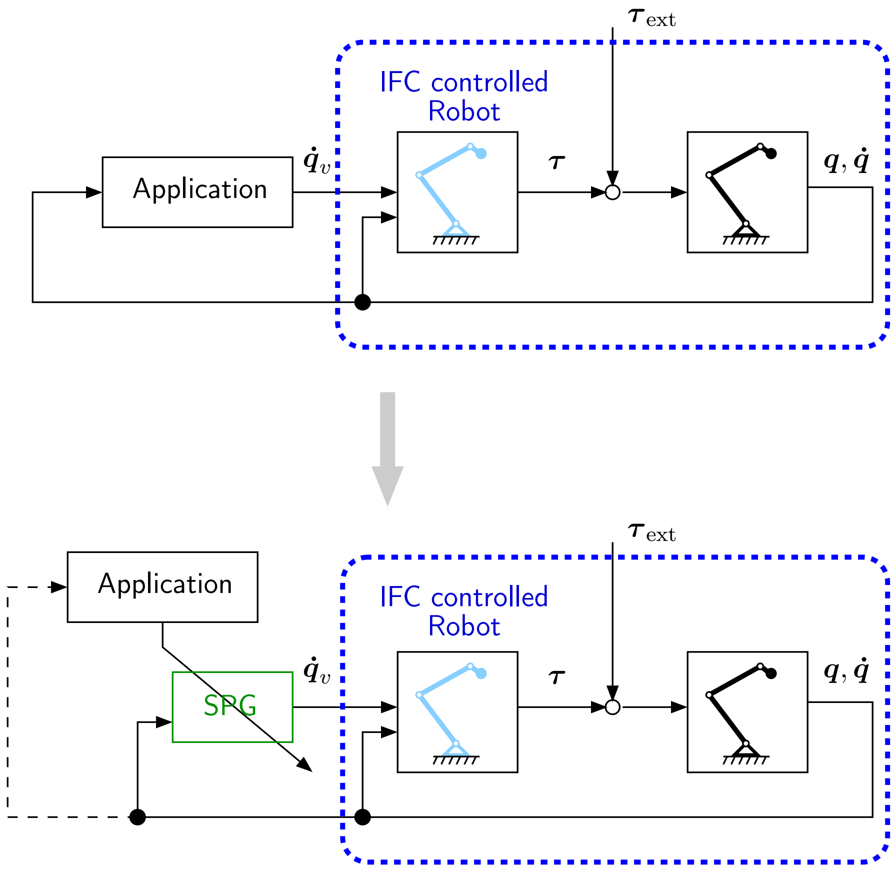


FIGURE 1.3. The SPG connected to a typical IFC loop. The application block is not necessarily part of the control loop anymore but has the function of parameterizing the SPG. However, the parameters can still be set in a dynamic way, incorporating the robots state or other sensory feedback. This potential feedback is indicated by the dotted line.

where a set of hierarchically ordered force and positioning subtasks on joint- or Cartesian-level are stated. This concept is similar to the *stack-of-tasks* framework in [73], where it is used to combine multiple tasks on a humanoid robot, like balancing, reaching and control of the field of view. The resulting structure has the following advantageous properties:

- Force and positioning equality and inequality tasks on joint- and Cartesian-level are captured with one unifying formalism.
- Intuitive, yet powerful task programming without requiring detailed information of the underlying structure, while the inherent compliance of the IFC is preserved.
- By treating the IFC-robot entity as a general abstraction layer and assuming no

additional external sensing, the developed approach is basically hardware independent as long as the robot features an IFC interface.

- By assuming a simple IFC-like control structure (2.20), this approach is in general also applicable for position controlled SEA robots, where a physical spring-damper system is present.
- As neither modification, nor detailed information of the underlying IFC is required, the proposed formalism is a suitable approach for dealing with so called closed architectures, where only limited access (i.e. an IFC interface) to the robot is granted.

1.3. Thesis Outline

The thesis is structured as follows. Chapter 2 covers briefly the related basics of robotics theory used in this thesis. In Chapter 3 a simplified method of regulating force and positioning tasks for indirect force controllers is derived and a first approach to execute several tasks simultaneously, using recursive nullspace mapping is investigated. In Chapter 4 the formalism is extended to support inequality tasks and a passivity based stabilization method is presented. In Chapter 5 the manipulation of constrained mechanisms with unknown geometry and unknown required interaction forces is treated. Chapter 6 finally summarizes the main results of the thesis and gives a brief outlook on future research directions.

Chapter 2

Robotics Background

This chapter gives a concise summary on the main robotics theory, which is required to follow the derivations in this thesis. For more details, it is referred to the relevant books instead, e.g. [76, 106].

2.1. Robot Kinematics

2.1.1. Position and Orientation Representation

The minimum number of coordinates required to locate a body in Euclidean space is six. Three for the position and three for the orientation. The position and orientation of a rigid body can be mathematically represented using *coordinate reference frames* or simply *frames*. A frame i consists of an origin O_i and an orthonormal base (three mutually orthogonal basis vectors in Euclidean space), which is fixed within a particular body (see Fig. 2.1).

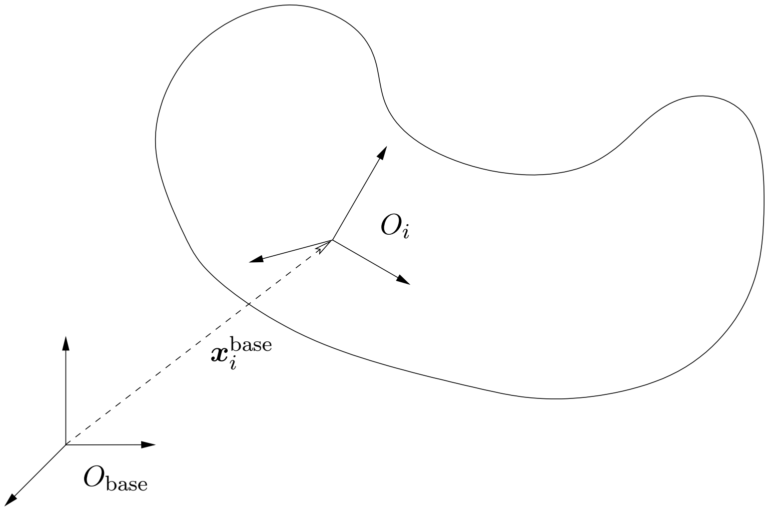


FIGURE 2.1. A rigid body in Euclidean space with associated frame.

The Cartesian pose of a frame can be mathematically represented by

$$\boldsymbol{x}_i^{\text{base}} = \begin{pmatrix} \boldsymbol{p}_i^{\text{base}} \\ \boldsymbol{o}_i^{\text{base}} \end{pmatrix}, \tag{2.1}$$

where $\boldsymbol{p}_i^{\text{base}} \in \mathbb{R}^3$ denotes the position and $\boldsymbol{o}_i^{\text{base}}$ the orientation of the frame i in the base frame. The unit and dimension of $\boldsymbol{o}$ depend on the chosen orientation representation. Multiple representations of the orientation exist, which are all advantageous in specific scenarios. As the actual representation is not relevant for the present work, they will not be covered here. For the sake of clarity, the indexes denoting the base and target frames are neglected for the most part of the document as it should be clear from the respective context.

2.1.2. Kinematic Chains/Trees

A robot can be described as a set of multiple rigid bodies (*links*), which are connected by moving *joints*. If the links are in a serial configuration this is referred to as a *kinematic chain*. Different kinematic structures can be composed of kinematic chains. The simplest is the *open kinematic chain*, when there is only one end-link and there are no additional constraints (see Fig. 2.2). This model is usually used to describe a robotic arm or *manipulator* in a non-contact state. In this context the final link is often called end-effector and its pose is denoted as $\boldsymbol{x}_{\text{ee}}$. The values determining the geometric relations between the link frames are called *kinematic parameters*. The number of joints n defines the *degree of freedom* (DoF) of the robot's configuration, which can be uniquely determined by a n-dimensional set of generalized coordinates $\boldsymbol{q}$. For revolute joints, $\boldsymbol{q}$ is a vector of joint angles. The pose of the end-effector, or in any other target frame with respect to the chain, can be computed from the generalized coordinates with the so called *forward kinematics* (FK) function

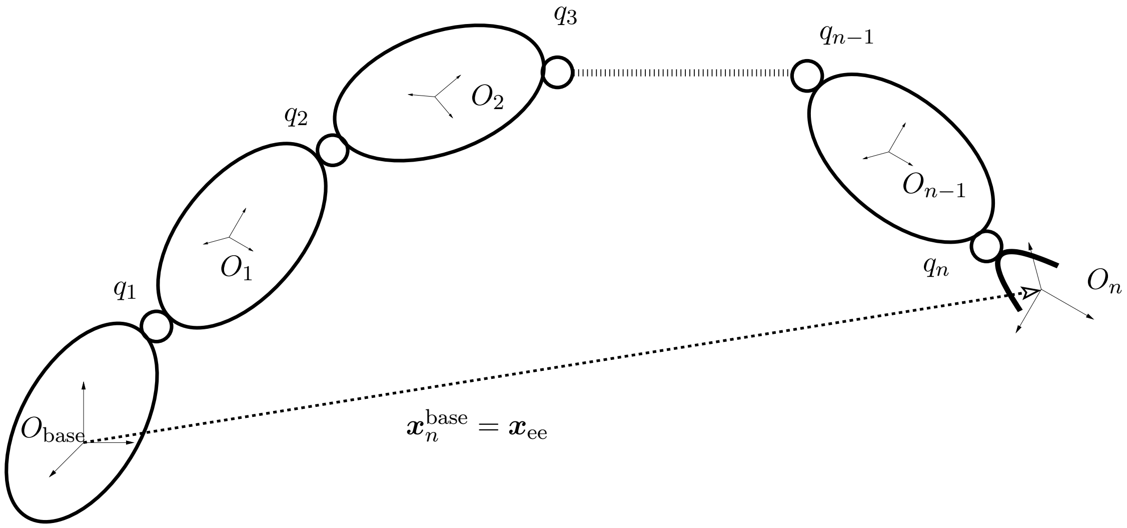


Figure 2.2. A robotic manipulator, modeled as an open kinematic chain with n joints and the final link as end-effector.

$$\boldsymbol{x} = g(\boldsymbol{q}). \tag{2.2}$$

Kinematic chains can be combined to a *kinematic tree* with more than one end-effector (see Fig. 2.3) and joints which belong to multiple chains at the same time. This means, that the motions of the different end-effectors are in general coupled as they (partially) need the same joints. In Chapter 3, techniques will be discussed, which show how tasks for different target frames can be decoupled. Such a structure is usually used to model a humanoid robot. The same formalism can be used to express the pose of multiple frames on one kinematic chain.

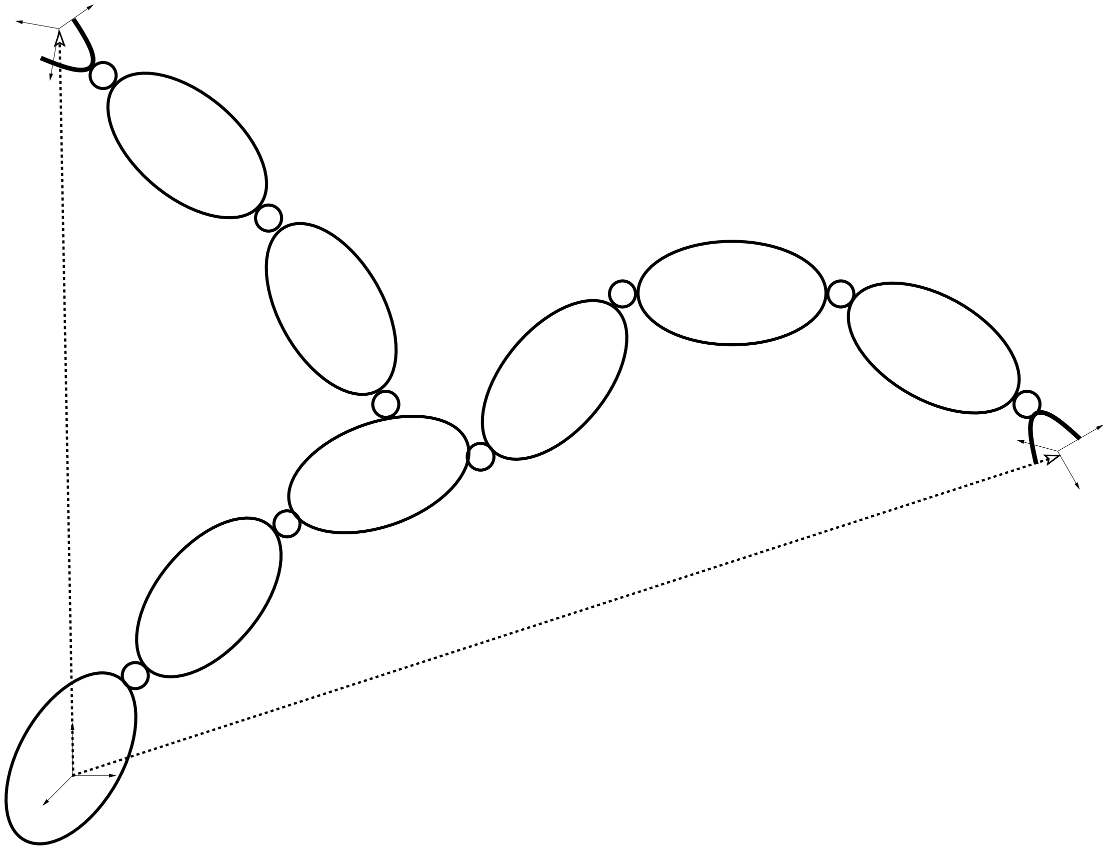


Figure 2.3. A dual arm manipulator, modeled as a kinematic tree with two end-effectors.

2.1.3. Redundancy

If $n > 6$ the robot is often denoted as redundant as there exist an infinite number of possible configurations, which result in the same Cartesian pose of the end-effector (see Fig. 2.4). A more general definition is, that a robot is redundant with respect to a m-dimensional task if $n > m$. For example, considering the task of holding a certain orientation (3-dimensional) a robot with four joints is redundant.

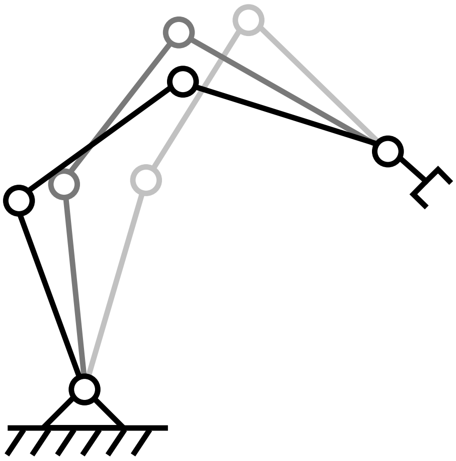


Figure 2.4. Redundancy of a planar manipulator with four independent joints. As the manipulator operates in a three-dimensional manifold it has one redundant DoF, hence the number of possible configurations for a given end-effector pose is infinite.

2.2. Inverse Kinematics and Motion Regulation

When regulating the robots motion, one is usually interested in controlling the motion of a particular Cartesian target frame on the robot, i.e. the end-effector. To obtain the associated joint-level commands one has to solve the *inverse kinematics* (IK) problem, which is to find the values of the joint positions, given the position and orientation of the target frame relative to the base and the kinematic parameters. For dealing with the infinite solutions due to redundancy, the most common approaches are based on a local solution to the IK problem, also known as *instantaneous inverse kinematics* (inst. IK). See Fig. 2.5 for the conceptual difference between the global and the local approach to motion regulation.

2.2.1. Jacobian

The common approach to obtain a local solution is to differentiate the FK map (2.2), which results in the $6 \times n$ *Jacobian matrix*

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial g(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{pmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \cdots \\ \frac{\partial x_2}{\partial q_1} & \frac{\partial x_2}{\partial q_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}. \tag{2.3}$$

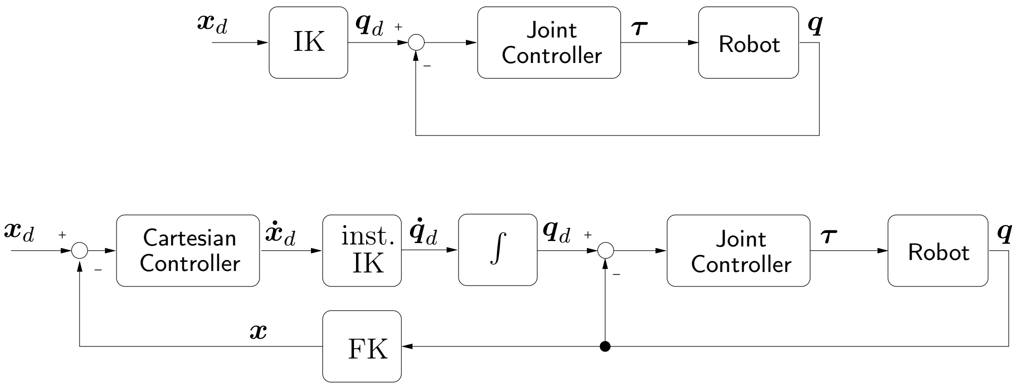


Figure 2.5. Conceptual difference between global joint space control (upper diagram) and instantaneous inverse kinematics control (lower diagram). In the global approach, the joint angles are computed directly from the desired Cartesian pose and fed into the joint-level controller, which computes the required joint torques or motor currents. In the local approach two controllers are involved. A Cartesian controller computes a desired twist $\dot{\boldsymbol{x}}_d$ depending on the Cartesian error, which is transformed to a desired joint velocity $\dot{\boldsymbol{q}}_d$ by applying inst. IK. Integration yields the desired joint position $\boldsymbol{q}_d$, which serves as input for the joint-level controller.

Differentiation with respect to time yields a set of equations of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}}, \tag{2.4}$$

with $\dot{\boldsymbol{x}}$ as a vector representing linear and angular velocity. One recognizes that $\boldsymbol{J}$ depends not only on the current configuration of the robot but also on the particular choice for pose representation. If the fixed angle orientation representation is chosen, then

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix}, \tag{2.5}$$

with $\boldsymbol{\omega}$ as the angular velocity of the target frame and $\boldsymbol{J}$ as the robot's base Jacobian with respect to that frame. $\dot{\boldsymbol{x}}$ is also referred to as twist or spatial velocity.

For a serial chain of six joints, the solution to the inst. IK problem for a six-dimensional motion command $\dot{\boldsymbol{x}}$ is then simply

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{x}} \tag{2.6}$$

for an invertible $\boldsymbol{J}$.

Manipulability The *manipulability* μ of a robot describes its ability to move freely in all directions in the workspace. Manipulability measures can be divided into two rough classes:

1. The ability to reach a certain position or set of positions
2. The ability to change the position or orientation at a given configuration

The first of these measures is directly related to the workspace of a manipulator. The second class of measures concerns the manipulability of a manipulator around a given configuration; that is, it is a local property.

To describe the local manipulability quantitatively, the determinant of the manipulator's Jacobian can be used:

$$\mu = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)} \tag{2.7}$$

General Task Jacobian A Jacobian can be computed for any target frame on the robot and for any m-dimensional generic task $\boldsymbol{\alpha}$ with respect to that frame, resulting in the general $m \times n$ *task Jacobian*

$$\boldsymbol{A}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{q}}. \tag{2.8}$$

A more general solution than (2.6), which is valid for any $\boldsymbol{A}$, is obtained by applying a generalized inverse $\boldsymbol{A}^{\#}$ to solve

$$\dot{\boldsymbol{q}} = \boldsymbol{A}^{\#}\dot{\boldsymbol{\alpha}}. \tag{2.9}$$

Here, often the *Moore-Penrose pseudo-inverse* $\boldsymbol{A}^{+}$ is used [10]. For $\text{rank}(\boldsymbol{A}) > m$,

$$\boldsymbol{A}^{+} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1} \tag{2.10}$$

obtains the *minimum-norm* solution of the under-determined problem in the redundant case. For $\text{rank}(\boldsymbol{A}) < m$,

$$\boldsymbol{A}^{+} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \tag{2.11}$$

obtains the *least-squares* solution of the over-determined case where the desired differential task variable can not be achieved.

Static Wrench Transmission The Jacobian can also be used to describe the relationship between the contact forces at a target frame and joint torques $\boldsymbol{\tau}$ via

$$\boldsymbol{J}^T \boldsymbol{h} = \boldsymbol{\tau} \tag{2.12}$$

$$\boldsymbol{J}\boldsymbol{J}^T \boldsymbol{h} = \boldsymbol{J}\boldsymbol{\tau} \tag{2.13}$$

$$\boldsymbol{h} = (\boldsymbol{J}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{\tau} = \boldsymbol{J}^{T+}\boldsymbol{\tau} \tag{2.14}$$

with $\boldsymbol{h}$ denoting the 6-dimensional contact wrench, a vector of forces $\boldsymbol{f}$ and moments $\boldsymbol{m}$. Relation (2.14) applies only to non-singular $\boldsymbol{J}$ as forces along singular directions are not reflected on the joint torques and are simply transfered into the mechanical structure.

2.2.2. Kinematic Singularities

A *singular configuration* of a robot is a configuration at which it looses the ability to achieve instantaneous motion in certain directions, called singular directions. A generic task $\boldsymbol{\alpha}$ is labeled singular if the associated task Jacobian $\boldsymbol{A}$ loses rank. Kinematic singularities can be classified into:

- *Boundary singularities* that occur when the manipulator is either outstretched or retracted. These singularities do not represent a true drawback, since they can be avoided if the manipulator is not driven to the boundaries of its reachable workspace. A typical example is the so called *elbow-lock*, which occurs if a robotic arm is completely outstretched.
- *Internal singularities* that occur inside the reachable workspace and are generally caused by the alignment of two or more axes of motion. Unlike the above, these singularities constitute a serious problem, as they can be encountered anywhere in the reachable workspace.

In Sec. 3.2.4 *algorithmic singularities* will also be discussed, which may occur if multiple tasks are pursuit in a hierarchical way.

To understand the problem which arises near a singularity, one has to take a closer look on the computation of the pseudo-inverse used in (2.9). A more efficient method than (2.10), respectively (2.11) involves the computation of the *Singular Value Decomposition* (SVD) of the task Jacobian $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U}$ is a $m \times m$ unitary matrix composed of column vectors $\boldsymbol{u}_i$ and $\boldsymbol{V}$ is a $n \times n$ unitary matrix composed of column vectors $\boldsymbol{v}_i$. $\boldsymbol{\Sigma}$ is a $m \times n$ block matrix with a leading $m \times m$ diagonal matrix containing the singular values $\sigma_i \geq 0$ of $\boldsymbol{A}$ with $i = 1, \ldots, m$ in decreasing order ($\sigma_i > \sigma_j$ for $i > j$), followed by $n - m$ zero columns. The pseudo-inverse is computed via

$$\boldsymbol{A}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{T} = \sum_{i=1}^{r} \frac{1}{\sigma_i} \boldsymbol{v}_i \boldsymbol{u}_i^T, \tag{2.15}$$

where $r \leq m$ is the rank of $\boldsymbol{A}$. Near singular configurations the task Jacobian becomes ill-conditioned with the smallest singular value σ_{min} approaching zero and hence $\frac{1}{\sigma_{\text{min}}} \to \infty$, hence the pseudo-inverse used in (2.9) will contain very high values, resulting in large joint velocities.

One way to cope with singularities is to avoid them in advance during task planning. In cases where this is not possible, as in reactive control schemes, proper measures need to be taken to traverse a singular configuration robustly. A popular approach to achieve this is the so called *damped pseudo-inverse* $\boldsymbol{A}^{+\lambda}$ based on the approximation

$$\frac{1}{\sigma_i} \approx \frac{\sigma_i}{\sigma_i^2 + \lambda^2}, \tag{2.16}$$

introducing the damping factor λ. The general approach is also referred to as *damped least squares* [77, 114, 71]. Replacing $\frac{1}{\sigma_i}$ in (2.15) by (2.16) results in

$$\boldsymbol{A}^{+\lambda} = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \boldsymbol{v}_i \boldsymbol{u}_i^T, \tag{2.17}$$

and introduces a regulated task error to grant a smooth transition through singular configurations. There are multiple ways for selecting λ. Usually σ_{min} is used to set λ dynamically:

$$\lambda^2 = \begin{cases} 0 & \text{when } \sigma_{\text{min}} \geq \sigma_{\text{crit}} \\ (1 - \frac{\sigma_{\text{min}}}{\sigma_{\text{crit}}})^2 \; \lambda_{\text{max}}^2 & \text{otherwise} \end{cases}, \tag{2.18}$$

where σ_{crit} is a threshold, which determines when the damping should start and λ_{max} is the maximal damping factor, which should be applied. An overview on different damping approaches can be looked up in [29]. The problem of the introduction of a task error has been recently tackled in [40].

2.3. Indirect Force Control

The dynamic equation of a robot controlled by the motor torque $\boldsymbol{\tau}$ is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_g = \boldsymbol{\tau} - \boldsymbol{\tau}_{\text{ext}} \tag{2.19}$$

with $\boldsymbol{M}$ as the joint inertia matrix, $\boldsymbol{c}$ as the effect of centrifugal and Coriolis forces, $\boldsymbol{\tau}_g$ as the torques due to gravity and $\boldsymbol{\tau}_{\text{ext}}$ as the external torques.

In contrast to motion or direct force control, the goal of IFC is not to regulate the position or interaction force but the relation between them. It is usually desired to achieve a compliant behavior by imposing some virtual mechanical relationship either at joint or Cartesian level. The various implementations depend on the concrete use case and the available hardware. In the following the two realizations relevant for this thesis are briefly summarized.

2.3.1. Joint-Level IFC

Assuming a torque interface, a simple stiffness controller can be directly implemented without any knowledge of the robot dynamics via

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g. \tag{2.20}$$

This corresponds basically to a PD controller with compensation of the gravitational torques. $\boldsymbol{K}$ and $\boldsymbol{D}$ are diagonal, positive definite stiffness and damping matrices.

If the dynamic parameters of the robot are known, (2.20) can be augmented with $\boldsymbol{\tau}_{\text{dyn}} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{c}$ to compensate for the dynamic effects of the robot's motion:

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g + \boldsymbol{\tau}_{\text{dyn}}, \tag{2.21}$$

resulting in a perfect impedance relation between position error and output torques.

With (2.20) or (2.21) each joint is emulating a mass-spring-damper system. Figure 2.6 visualizes the relation between set-point $\boldsymbol{q}_v$ and physical robot $\boldsymbol{q}$ for one joint. Due to the impedance-like behavior of the robot (motion in – force out), such schemes are also referred to as joint space impedance control.

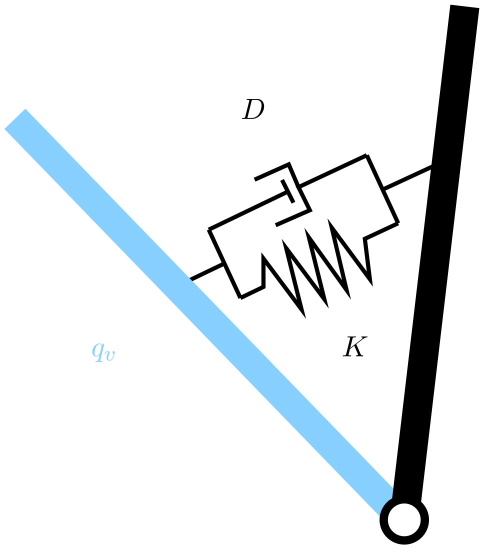


Figure 2.6. IFC-induced joint behavior for one joint.

2.3.2. Admittance Control

The higher the stiffness in pure impedance control, the smaller the pose error will be. On the other hand, a high stiffness leads to large contact forces. A solution is to separate motion control from impedance control as follows. An impedance law in the outer loop computes the deviation from the desired trajectory $\boldsymbol{x}_d$ according to

$$\boldsymbol{h} = \boldsymbol{M}_x \Delta\ddot{\boldsymbol{x}} + \boldsymbol{D}_x \Delta\dot{\boldsymbol{x}} + \boldsymbol{K}_x \Delta\boldsymbol{x}, \tag{2.22}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x}_c - \boldsymbol{x}_d$, with $\boldsymbol{x}_c$ as the compliant trajectory of equilibrium poses which is fed to an inner motion control loop. The 6×6 matrices $\boldsymbol{M}_x$, $\boldsymbol{D}_x$ and $\boldsymbol{K}_x = \mathrm{diag}(\boldsymbol{K}_p,\ \boldsymbol{K}_o)$ is the Cartesian inertia, damping and stiffness matrix describing the desired target impedance. They are usually constant and diagonal.

Using the contact force as input, equation (2.22) can be solved for $\Delta\boldsymbol{x}$, $\Delta\dot{\boldsymbol{x}}$ and $\Delta\ddot{\boldsymbol{x}}$ and executed on the manipulator by any motion controller. The impedance part of the control law modifies the desired trajectory according to the measured wrench, hence it only comes into effect during contact with the environment. In free space the measured wrench, and therefore $\Delta\boldsymbol{x}$ is zero, so the desired trajectory remains unaffected. This control method is known as Cartesian position based *admittance control*, due to the equivalence to a mechanical admittance with force as input and motion as output quantities. It is straight forward to formulate and implement an admittance controller in joint space, provided joint torque measurements are available.

A force following behavior can be achieved by emulating only a mass-damper system, thus an admittance with zero stiffness matrix ($\boldsymbol{K}_x = \boldsymbol{0}$):

$$\boldsymbol{h} = \boldsymbol{M}_x \ddot{\boldsymbol{x}} + \boldsymbol{D}_x \dot{\boldsymbol{x}}. \tag{2.23}$$

Fig. 2.7 shows the control scheme of position based admittance control.

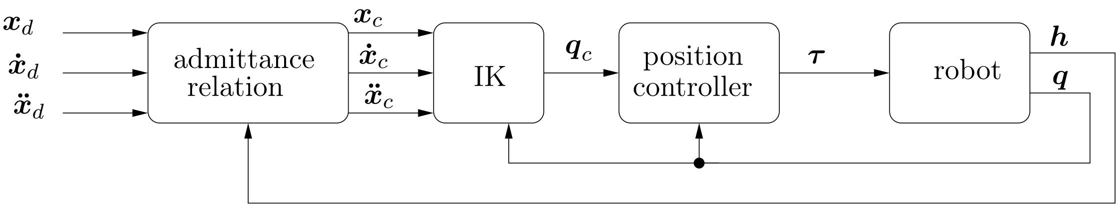


Figure 2.7. Position based Cartesian admittance control loop.

Joint- vs. Cartesian Compliance A notable property of IFC's realizing joint compliance as (2.20) is, that the Cartesian position deviation and the applied interaction forces are not aligned. This is often neglected by application programmers using joint-level IFC's. Fig. 2.8 depicts this misalignment.

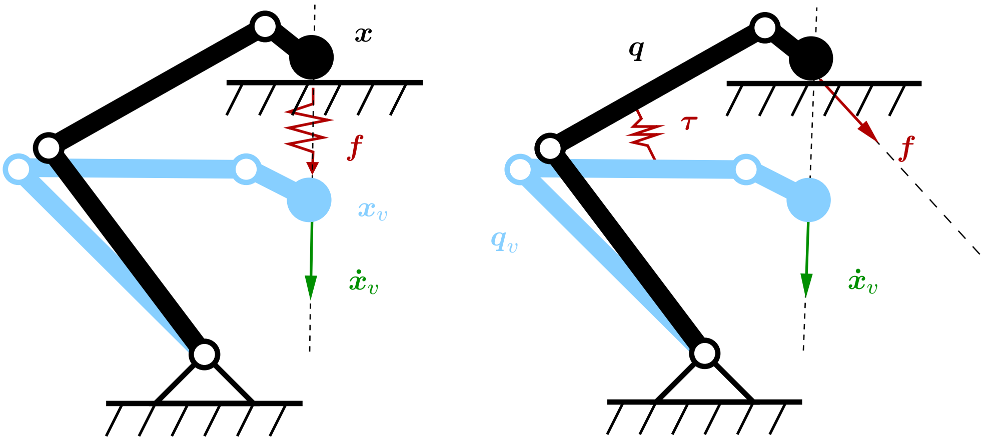


Figure 2.8. Comparison between joint- and Cartesian-level IFC. The position difference $\boldsymbol{x}_v - \boldsymbol{x}$ is in general not aligned with the applied interaction wrench for joint-level IFC's.

Chapter 3

Set-Point Selection for Force and Position Regulation

Commanding a set-point to an indirect force controlled robot affects its position as well as the applied interaction forces. Therefore, decoupling force and position commands is not trivial for joint space IFC's. A common example is controlling the contact force and the orientation of the end-effector simultaneously in a surface tracking task.

In this chapter a possible approach to achieve this is derived. First in Sec. 3.1, force and positioning tasks in joint and Cartesian space are generalized in the context of IFC. Simplifying assumptions are defined, which trade accuracy for generalization of force and position regulation while keeping the inherent compliance of the indirect force controlled robot. Second, standard techniques for executing multiple tasks simultaneously are applied, using the previously derived formalism from section 3.2. The experiments are summarized in section 3.3 and section 3.4 concludes the chapter.

3.1. Generalized Force and Position Regulation in IFC Context

3.1.1. Task Variables

Consider a pure IFC interface like (2.20). Neither position $\boldsymbol{q}$ nor the full interaction torque $\boldsymbol{\tau}$ can be regulated directly, hence the following replacement task variables are declared, considering (2.2) and (2.14):

- $\boldsymbol{q}_v$ – virtual joint space position
- $\boldsymbol{x}_v = \boldsymbol{g}(\boldsymbol{q}_v)$ – virtual Cartesian pose (i.e. end-effector)
- $\boldsymbol{\tau}_s = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q})$ – static interaction joint torques
- $\boldsymbol{h}_s = \boldsymbol{J}^{+T}\boldsymbol{\tau}_s = \boldsymbol{J}^{+T}\boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q})$ – static interaction wrench

These variables have a close relation to the actual variables of interest ($\boldsymbol{q}, \boldsymbol{x}, \boldsymbol{\tau}$, $\boldsymbol{h}$) but in contrast to them, they can be directly related to $\boldsymbol{q}_v$. Naturally, in addition to these archetypes, any task variable can be defined, which is directly related to $\boldsymbol{q}_v$, e.g. any point on the robot with it's given Jacobian $\boldsymbol{J}_c$. In the following, the impact on the task error when regulating these replacement variables is discussed.

3.1.1.1. Joint Position Error

For no external torques ($\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{0}$), the manipulators equilibrium configuration $\boldsymbol{q}_{\text{equ}}$ is equal to the specified set-point $\boldsymbol{q}_v$, resulting in the steady state error $\tilde{\boldsymbol{q}} = \boldsymbol{q}_v - \boldsymbol{q}_{\text{equ}} = \boldsymbol{0}$. If an external torque is present, respectively $\boldsymbol{q}_v$ can not be reached due to an obstacle, the steady state error can be derived from the static relation $\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}_{\text{equ}})$, hence

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_v - \boldsymbol{q}_{\text{equ}} = \boldsymbol{K}^{-1}\boldsymbol{\tau}_{\text{ext}} \tag{3.1}$$

The tracking performance depends mainly on the actual IFC implementation and its parameters (mainly $\boldsymbol{K}$) and is characterized by the well known trade-off between accuracy and compliance.

3.1.1.2. Cartesian Error

The steady state Cartesian error $\tilde{\boldsymbol{x}}$ can be directly derived from the joint space error via the forward kinematics (2.2)

$$\tilde{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{q}_v) - \boldsymbol{g}(\boldsymbol{q}_{\text{equ}}) \tag{3.2}$$

and depends, like the joint space error, on the concrete IFC implementation but also on the actual configuration of the robot.

3.1.1.3. Wrench/torque error

Considering only the static components of interaction torques/wrenches is an acceptable simplification for low velocities and leads to the steady state errors

$$\tilde{\boldsymbol{\tau}} = \boldsymbol{0} \tag{3.3}$$

$$\tilde{\boldsymbol{h}} = \boldsymbol{0} \tag{3.4}$$

assuming perfect compensation of the gravitational torques and a static equilibrium configuration. If the robot is moving, the uncompensated robot dynamics and the damping forces from the IFC lead to a force/torque error, which increases with the motion speed. The actual error depends mainly on the quality of the IFC and the damping parameter $\boldsymbol{D}$ in (2.20). However, for moderate speeds the static components of the force dominate clearly, therefore this error is accepted for the sake of safe and compliant interaction.

3.1.2. Generalized Task Regulation

Robot motion control can be roughly categorized into global and local approaches. Global approaches are used to obtain a complete trajectory. They are computationally expensive and are usually applied in offline motion planning methods. The advantage is that the whole trajectory is known in advance. The question of how to treat a dynamic environment (i.e. obstacle avoidance) and achieve a reactive behavior is subject of different research directions, e.g. [89] and the resulting *elastic strips* framework from Brock et. al. [14].

Local approaches on the other hand, regard only the current state of the robot and the eventually sensory input to compute a differential command. They are computationally far less expensive than the global approaches and are usually applied in online trajectory generation schemes, where the evolution of the task variable is continuously adapted based on sensory feedback information. However, it is a well known fact, that local approaches have the disadvantage that they might get stuck in certain configurations (local minima) when navigating through obstacles, although the existence of a valid path. Nevertheless, they allow an easy integration of multiple prioritized tasks and hence are the most common approach if the robot is redundant.

Local approaches apply the inversion of the differential relation between the task and the actuating variable ($\boldsymbol{q}_v$), captured in the so called task Jacobian $\boldsymbol{A}$. In the following, $\boldsymbol{A}$ is derived for the four task archetypes.

3.1.2.1. Deriving the Task Jacobians

Joint Position For the joint position $\boldsymbol{q}_v$ the task Jacobian is trivial:

$$\dot{\boldsymbol{q}}_v = \boldsymbol{I}_n \dot{\boldsymbol{q}}_v \tag{3.5}$$

$$\boldsymbol{A}_q = \boldsymbol{I}_n, \tag{3.6}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix.

Cartesian Pose For the Cartesian pose, the instantaneous kinematics relation (2.4) is used to obtain

$$\dot{\boldsymbol{x}}_v = \boldsymbol{J}_v \dot{\boldsymbol{q}}_v \tag{3.7}$$

$$\boldsymbol{A}_x = \boldsymbol{J}_v, \tag{3.8}$$

with $\boldsymbol{J}_v = \boldsymbol{J}(\boldsymbol{q}_v)$ denoting the Jacobian of the virtual manipulator.

Joint Torque Taking the time derivative of $\boldsymbol{\tau}_s$ leads to

$$\dot{\boldsymbol{\tau}}_s = \boldsymbol{K}(\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{q}}) \tag{3.9}$$

$$\dot{\boldsymbol{\tau}}_s + \boldsymbol{K}\dot{\boldsymbol{q}} = \boldsymbol{K}\dot{\boldsymbol{q}}_v. \tag{3.10}$$

It can be noted that in this case, the instantaneous relation between the task variable $\boldsymbol{\tau}_s$ and $\boldsymbol{q}_v$ depends also on the joint velocity of the physical robot $\dot{\boldsymbol{q}}$. Therefore, to obtain a plain linear relation, similar to (3.5) and (3.7), the differential task variable has to be compensated for the effects of $\dot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{\beta}}_\tau = \dot{\boldsymbol{\tau}}_s + \boldsymbol{K}\dot{\boldsymbol{q}} \tag{3.11}$$

$$\boldsymbol{A}_\tau = \boldsymbol{K} \tag{3.12}$$

$$\dot{\boldsymbol{\beta}}_\tau = \boldsymbol{A}_\tau \dot{\boldsymbol{q}}_v, \tag{3.13}$$

where $\dot{\boldsymbol{\beta}}_\tau$ is the compensated differential task variable.

Wrench Taking the time derivative of (2.14) leads to

$$\dot{\boldsymbol{h}}_s = \dot{\boldsymbol{J}}^{T+}\boldsymbol{\tau}_s + \boldsymbol{J}^{T+}\dot{\boldsymbol{\tau}}_s. \tag{3.14}$$

By assuming only moderate joint velocities, the term with the derivative of the Jacobian is neglected, resulting in the simplified relation

$$\dot{\boldsymbol{h}}_s = \boldsymbol{J}^{T+}\dot{\boldsymbol{\tau}}_s = \boldsymbol{J}^{T+}\boldsymbol{K}(\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{q}}) \tag{3.15}$$

$$\dot{\boldsymbol{h}}_s + \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}} = \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}}_v \tag{3.16}$$

With this, the wrench task Jacobian $\boldsymbol{A}_h$ and the compensated differential wrench task variable $\dot{\boldsymbol{\beta}}_h$ can be stated as

$$\dot{\boldsymbol{\beta}}_h = \dot{\boldsymbol{h}}_s + \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}} \tag{3.17}$$

$$\boldsymbol{A}_h = \boldsymbol{J}^{T+}\boldsymbol{K} \tag{3.18}$$

$$\dot{\boldsymbol{\beta}}_h = \boldsymbol{A}_h \dot{\boldsymbol{q}}_v \tag{3.19}$$

This relation is obviously only valid for Jacobians, which are sufficiently far away from singularities. Due to the inversion in (3.18), in the proximity of a singular configuration $\dot{\boldsymbol{\beta}}_h$ gets oversensitive to small changes in $\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{q}}$. It has also to be noted, that if (3.18) is used for redundancy resolution, the resulting torques could contain null-space components, which would result in an uncontrolled motion and deform the wrench-task. Therefore, the dynamically consistent pseudo-inverse $(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1})$ should be used instead of $\boldsymbol{J}^+$ to cancel potential null-space torques.

To unify the four task archetypes, a general compensated differential task variable $\dot{\boldsymbol{\beta}}$ is defined, so that together with the general m-dimensional task $\boldsymbol{\alpha}$ and general $m \times n$ task

Jacobian $\boldsymbol{A}$ the following relations can be stated:

$$\dot{\boldsymbol{\beta}} = \dot{\boldsymbol{\alpha}} + \gamma(\dot{\boldsymbol{q}}) \tag{3.20}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{A}\dot{\boldsymbol{q}}_v, \tag{3.21}$$

where $\boldsymbol{\gamma}(\dot{\boldsymbol{q}})$ is the respective compensation for $\dot{\boldsymbol{q}}$.

Table 3.1 summarizes the four basic task types with associated task Jacobian and joint velocity compensation $\boldsymbol{\gamma}$.

TABLE 3.1. The four basic task types

$\boldsymbol{\alpha}$	$\boldsymbol{q}_v$	$\boldsymbol{x}_v$	$\boldsymbol{\tau}_s$	$\boldsymbol{h}_s$
γ	$\boldsymbol{0}$	$\boldsymbol{0}$	$\boldsymbol{K}\dot{\boldsymbol{q}}$	$\boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}}$
$\boldsymbol{A}$	$\boldsymbol{I}_n$	$\boldsymbol{J}_v$	$\boldsymbol{K}$	$\boldsymbol{J}^{T+}\boldsymbol{K}$

3.1.2.2. Trimmed Task Space

Tasks do not have to be necessarily defined in the full task space. Often only a particular subspace is relevant, so that the released degrees of freedom can be used to achieve other tasks, what will be treated in Sec. 3.2. The base of this subspace can be described by a set of orthonormal vectors, which form the columns of a matrix $\boldsymbol{S}$. With $\boldsymbol{\alpha}$ being defined in the subspace coordinates, the basic task Jacobians from table 3.1 have to be transformed to the subspace coordinates via

$$\boldsymbol{A} := \boldsymbol{S}^T\boldsymbol{A}. \tag{3.22}$$

Examples of tasks, which occupy only a part of the work- respectively joint space are depicted in Fig. 3.1. $\boldsymbol{S}$ serves two purposes here. First, it can be used as a selection matrix to determine the relevant directions of the full task space, similar to the selection matrix in hybrid force and position control [90]. For example, controlling only the

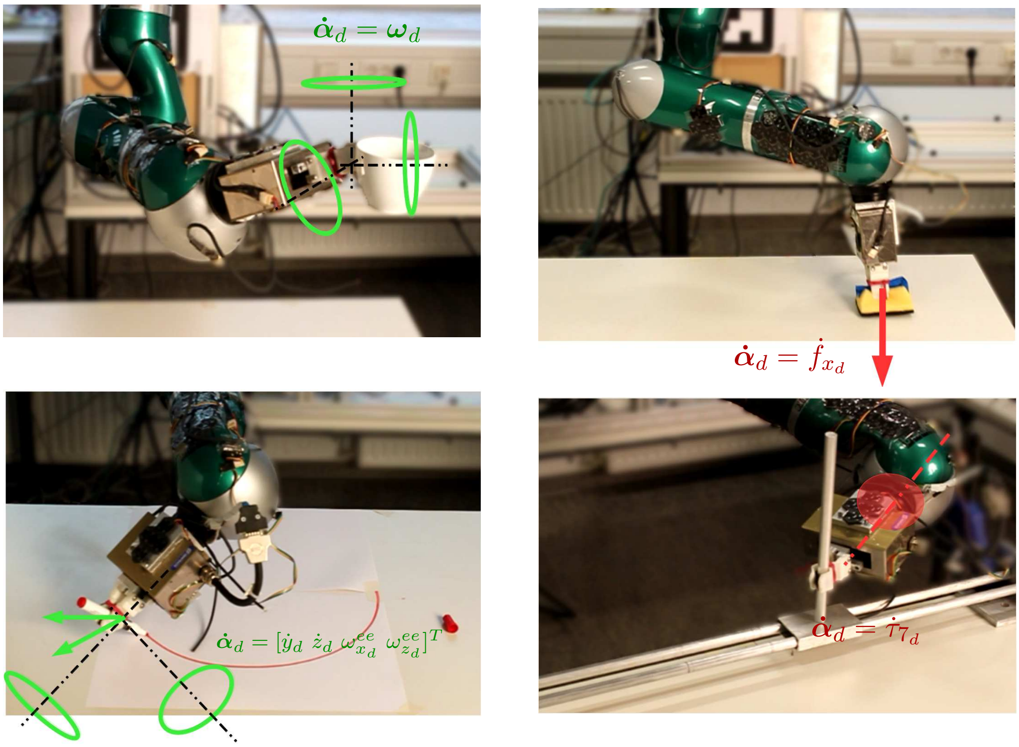

$\dot{\boldsymbol{\alpha}}_d = \boldsymbol{\omega}_d$
$\dot{\boldsymbol{\alpha}}_d = \dot{f}_{x_d}$
$\dot{\boldsymbol{\alpha}}_d = [\dot{y}_d\ \dot{z}_d\ \omega^{ee}_{x_d}\ \omega^{ee}_{z_d}]^T$
$\dot{\boldsymbol{\alpha}}_d = \dot{\tau}_{7_d}$

Figure 3.1. Some sample tasks which occupy only a part of the Cartesian/joint space, i.e. keeping a constant orientation (a), apply a certain contact force (b), drawing on a plain surface while relaxing invariant control directions (c), manipulating a mechanism while minimizing the torques on a certain joint (d).

rotational degrees of freedom with

$$\boldsymbol{S}_\omega = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

or only the force or motion along the x-direction

$$\boldsymbol{S}_x = [1\ 0\ 0\ 0\ 0\ 0\ 0]^T. \tag{3.24}$$

Second, it can be used to formulate the task (partially) in a convenient coordinate system, e.g. the end-effector frame. For circle drawing example in Fig. 3.1c, where the manipulator has to follow a trajectory in the base $y-z-$ plane, while holding a constant angle around the end-effector x and z axis:

$$\boldsymbol{S}_{\omega_{ee}} = \mathrm{diag}(\mathbf{0}_3, \boldsymbol{R}_{ee}) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

where $\boldsymbol{R}_{ee}$ is the 3×3 rotation matrix of the end-effector. It follows that $\boldsymbol{S}$ is either $6 \times m$ or $n \times m$, depending on whether the task is defined in Cartesian or in joint space.

3.1.2.3. Generalized Controller

The goal of the regulation is to bring the task variable $\boldsymbol{\alpha}$ continuously to a desired state $\boldsymbol{\alpha}_d$. The simplest way to achieve this is by using the task error

$$\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_d - \boldsymbol{\alpha} \tag{3.26}$$

to formulate a general proportional task-level controller

$$\dot{\boldsymbol{\alpha}}_d = \boldsymbol{\Lambda}\tilde{\boldsymbol{\alpha}} + \dot{\boldsymbol{\alpha}}_{\mathrm{ff}}. \tag{3.27}$$

$\dot{\boldsymbol{\alpha}}_{\mathrm{ff}}$ denotes an optional feedforward term and $\boldsymbol{\Lambda}$ a diagonal gain matrix, which tunes the convergence rate of this simple regulation. The main purpose of $\dot{\boldsymbol{\alpha}}_{\mathrm{ff}}$ is to allow a direct specification of the differential task variable, for example when implementing velocity fields for obstacle avoidance or tracking. The desired compensated task variable is then

$$\dot{\boldsymbol{\beta}}_d = \dot{\boldsymbol{\alpha}}_d + \boldsymbol{\gamma}(\dot{\boldsymbol{q}}) \tag{3.28}$$

To obtain the associated velocity input $\dot{\boldsymbol{q}}_{\boldsymbol{v}}$ for the IFC, (3.21) is solved for $\dot{\boldsymbol{q}}_{\boldsymbol{v}}$ by applying the pseudo-inverse:

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = \boldsymbol{A}^{+}\dot{\boldsymbol{\beta}}_d. \tag{3.29}$$

3.2. Simultaneous Force and Position Control for Indirect Force Controlled Robots

With the generalized task regulation, the original problem of proper task separation can finally be tackled. Treatment of multiple tasks can be basically approached in two ways. The first is by assigning different weights to the usually concurring tasks. The second is strict separation of tasks in a hierarchical way. The weighting strategy requires additional tuning of the weights and subtasks are not separated cleanly. Therefore the second method is often preferred. Combining multiple tasks in a hierarchical manner dates back to [65], where it has been done for kinematic control using nullspace mapping to resolve the manipulator's kinematic redundancy. The basic concepts have been used and expanded since then in many publications e.g. [70, 78, 57, 105, 6, 101]. A survey of different methods can be found in [79]. The focus lies mainly on redundancy resolution, without regarding lower dimensional subtasks. Most of the works are limited to kinematic or force control only.

3.2.1. Nullspace Projection

A linear map $L : V \to W$ between two vector spaces V and W represented by an $m \times n$ matrix $\boldsymbol{L}$ gives

$$\boldsymbol{w} = \boldsymbol{L}\boldsymbol{v}, \tag{3.30}$$

where $\boldsymbol{v}$ and $\boldsymbol{w}$ are elements of $\boldsymbol{V}$ and $\boldsymbol{W}$, in linear algebra and functional analysis, the nullspace $\ker(L)$ or kernel of L is the set of all elements $\boldsymbol{v}_N$ of V for which $L(\boldsymbol{v}_N) = \mathbf{0}_w$, where $\mathbf{0}_w$ denotes the zero vector in W. Hence the set of solutions to the equation

$$\boldsymbol{L}\boldsymbol{v} = \mathbf{0} \tag{3.31}$$

Putting this in the context of instantaneous robot control, the nullspace of the linear map (3.21) is the set of virtual joint velocity vectors, which have no instantaneous impact on the differential objective $\dot{\boldsymbol{\beta}}$. For $n > m$ the nullspace is not trivial and can be used to track secondary objectives without violating the main one. To realize this, the orthogonal projection [64] of a vector into the nullspace of L is defined as

$$\boldsymbol{N}(\boldsymbol{L}) = \boldsymbol{I}_n - \boldsymbol{L}^{+}\boldsymbol{L}, \tag{3.32}$$

mapping a generic vector in V to $\ker(L)$ so that

$$\boldsymbol{L}\boldsymbol{N}(\boldsymbol{L})\boldsymbol{v} = \boldsymbol{0} \tag{3.33}$$

Fig. 3.2 depicts this.

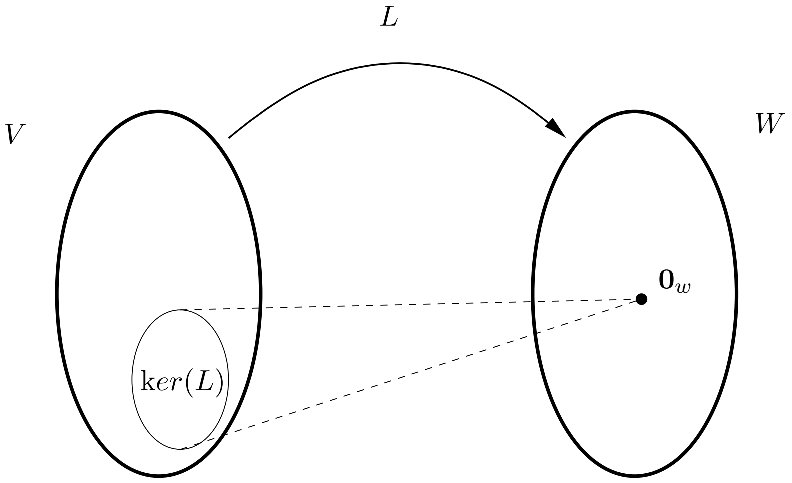


FIGURE 3.2. Visualization of a general linear map L between the two vector spaces V and W with nullspace or kernel $\ker(L)$.

3.2.1.1. Adding a Joint Nullspace Task

By applying nullspace projection, (3.29) can be extended to a more general solution to the instantaneous inverse problem:

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = \boldsymbol{A}^{+}\dot{\boldsymbol{\beta}}_d + \boldsymbol{N}(\boldsymbol{A})\dot{\boldsymbol{q}}_{v_N}, \tag{3.34}$$

where $\dot{\boldsymbol{q}}_{v_N}$ is an arbitrary joint velocity vector to be executed without violating $\dot{\boldsymbol{\beta}}_d$. Fig. 3.3 depicts this exemplary for the 2 DoF case. In classical redundancy resolution for $n > 6$-DoF robots, (3.34) is used with proper design of $\dot{\boldsymbol{q}}_{v_N}$ to maximize the distance of the joint limits or more general, to minimize a position dependent scalar cost function $p(\boldsymbol{q})$, like proposed in [64], where an approach to avoid joint limits was demonstrated by defining

$$p = \sum_{i=1}^{n} \left(\frac{q_i - \bar{q}_i}{\bar{q}_i - q_{i_M}} \right)^2 \tag{3.35}$$

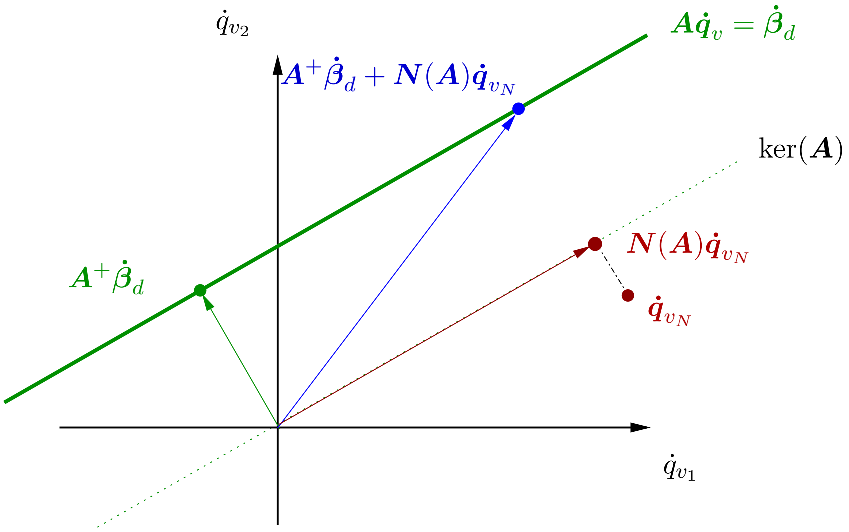

$\dot{q}_{v2}$
$\boldsymbol{A}\dot{\boldsymbol{q}}_v = \dot{\boldsymbol{\beta}}_d$
$\boldsymbol{A}^+\dot{\boldsymbol{\beta}}_d + \boldsymbol{N}(\boldsymbol{A})\dot{\boldsymbol{q}}_{v_N}$
$\ker(\boldsymbol{A})$
$\boldsymbol{N}(\boldsymbol{A})\dot{\boldsymbol{q}}_{v_N}$
$\boldsymbol{A}^+\dot{\boldsymbol{\beta}}_d$
$\dot{\boldsymbol{q}}_{v_N}$
$\dot{q}_{v1}$

Figure 3.3. Visualization of the solution obtained by (3.34) for 2 degrees of freedom. The green line represents the set of solutions for a desired $\dot{\boldsymbol{\beta}}_d$ and the green arrow the associated least norm solution. The green dotted line represents the nullspace of $\boldsymbol{A}$ and the red arrow the mapping of $\dot{\boldsymbol{q}}_{v_N}$ on it. The blue arrow is the sum of the green and red component and represents the final solution, which fulfills $\dot{\boldsymbol{q}}_{v_N}$ as good as possible without violating $\dot{\boldsymbol{\beta}}_d$.

and applying

$$\dot{\boldsymbol{q}}_{v_N} = k\frac{\partial p}{\partial \boldsymbol{q}}, \tag{3.36}$$

where i is the joint index, q_{i_m} and q_{i_M} are the joint limits and $\bar{q}_i = (q_{i_m} + q_{i_M})/2$ is the middle of the joint's range. k serves as a constant positive gain.

3.2.1.2. Generalized Nullspace Tasks

To combine different task types, a generalization of 3.34 is required. For better understanding of the following, a set of two objectives described by the differential task variables $\dot{\boldsymbol{\beta}}_1$ and $\dot{\boldsymbol{\beta}}_2$ with the associated task Jacobians $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ and nullspace projection matrix $\boldsymbol{N}_1 = \boldsymbol{N}(\boldsymbol{A}_1)$ can be defined. The priorities of the tasks follow the increasing numerical order.

The straight forward solution is to compute the required joint velocities for the second task via the pseudo-inverse and map the resulting joint velocities to the nullspace of the

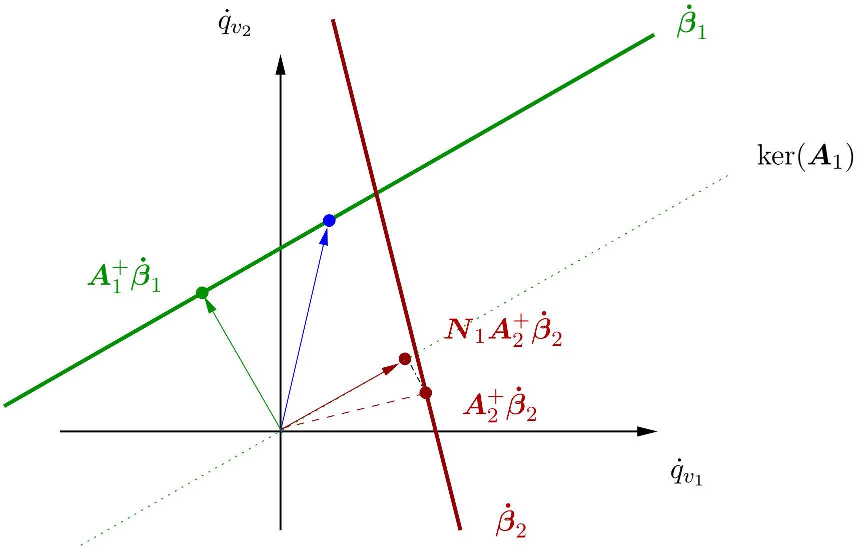

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_1$
$\ker(\boldsymbol{A}_1)$
$\boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1$
$\boldsymbol{N}_1\boldsymbol{A}_2^+\dot{\boldsymbol{\beta}}_2$
$\boldsymbol{A}_2^+\dot{\boldsymbol{\beta}}_2$
$\dot{q}_{v_1}$
$\dot{\boldsymbol{\beta}}_2$

FIGURE 3.4. Visualization of the solution obtained by (3.37) for 2 degrees of freedom. The red line represents the set of solutions for a second, lower priority task $\dot{\boldsymbol{\beta}}_2$ and the dashed red line, the associated least norm solution. The red arrow represents the mapping of this solution to the nullspace of the first task $\dot{\boldsymbol{\beta}}_1$. The resulting solution (blue arrow) in general does not state the optimal outcome, which is the intersection between the red and green line.

first task as proposed in [19] for Cartesian velocities.

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = \boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1 + \boldsymbol{N}_1\boldsymbol{A}_2^+\dot{\boldsymbol{\beta}}_2 \tag{3.37}$$

This approach is depicted in Fig. 3.4, where one can immediately see that applying the nullspace mapping after computing the joint velocities for the second task does not result in the optimal solution, i.e. the common solution is not determined even though it exists. The introduced error to the second task depends on its extent of coupling to the first one. One must admit however, that this solution is robust to so called algorithmic singularities, which will be treated later in this section.

To obtain the common solution one has to:

1. Postmultiply $\boldsymbol{A}_2$ with $\boldsymbol{N}_1$ to incorporate the nullspace mapping before the inversion.
2. Compensate for effects of $\dot{\boldsymbol{\beta}}_1$ on $\dot{\boldsymbol{\beta}}_2$.

The resulting solution

$$\dot{\boldsymbol{q}}_{\boldsymbol{v_1}} = \boldsymbol{A}_1^+ \dot{\boldsymbol{\beta}}_1 \tag{3.38}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = \dot{\boldsymbol{q}}_{\boldsymbol{v_1}} + (\boldsymbol{A}_2\boldsymbol{N}_1)^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v_1}}), \tag{3.39}$$

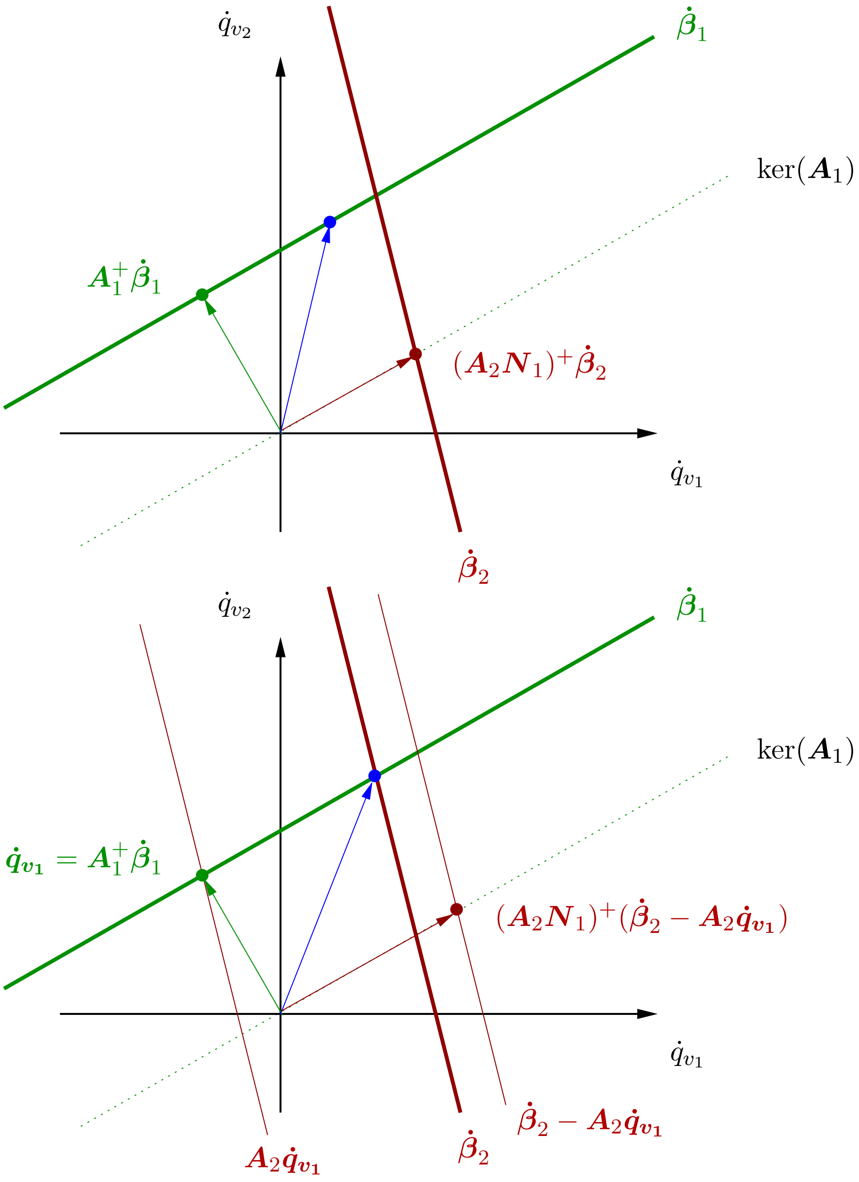

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_1$
$\ker(\boldsymbol{A}_1)$
$\boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1$
$(\boldsymbol{A}_2\boldsymbol{N}_1)^+\dot{\boldsymbol{\beta}}_2$
$\dot{q}_{v_1}$
$\dot{\boldsymbol{\beta}}_2$
$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_1$
$\ker(\boldsymbol{A}_1)$
$\dot{\boldsymbol{q}}_{\boldsymbol{v_1}} = \boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1$
$(\boldsymbol{A}_2\boldsymbol{N}_1)^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v_1}})$
$\dot{q}_{v_1}$
$\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v_1}}$
$\boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v_1}}$
$\dot{\boldsymbol{\beta}}_2$

Figure 3.5. Solution obtained by (3.39). In the upper figure, the nullspace mapping is incorporated before the inversion of $\boldsymbol{A}_1$, realized by modification of the second task Jacobian $\boldsymbol{A}_2$. The final solution is depicted in the lower figure, where in addition the task variable $\dot{\boldsymbol{\beta}}_2$ is compensated for the effects of the higher priority task $\dot{\boldsymbol{\beta}}_1$.

was firstly proposed in [43] to combine two Cartesian positioning tasks. A visualization is given in Fig. 3.5. This solution however, is vulnerable to algorithmic singularities.

3.2.2. Extension to Multiple Objectives

To discuss the generalization of the approach to multiple tasks, a third objective $\dot{\boldsymbol{\beta}}_3$ is introduced, which should be executed together with $\dot{\boldsymbol{\beta}}_1$ and $\dot{\boldsymbol{\beta}}_2$. Also the associated task Jacobian $\boldsymbol{A}_3$ and nullspace projection matrix for the second task $\boldsymbol{N}_2 = \boldsymbol{N}(\boldsymbol{A}_2)$ are defined.

To obtain a solution for multiple tasks, a straight forward approach is to compute the solution for $\dot{\boldsymbol{\beta}}_1$ and $\dot{\boldsymbol{\beta}}_2$ using (3.37) as proposed in [19].

$$\begin{aligned}\dot{\boldsymbol{q}}_{\boldsymbol{v}} &= \boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1 + \boldsymbol{N}_1(\boldsymbol{A}_2^+\dot{\boldsymbol{\beta}}_2 + \boldsymbol{N}_2\boldsymbol{A}_3^+\dot{\boldsymbol{\beta}}_3) \\ &= \boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1 + \boldsymbol{N}_1\boldsymbol{A}_2^+\dot{\boldsymbol{\beta}}_2 + \boldsymbol{N}_1\boldsymbol{N}_2\boldsymbol{A}_3^+\dot{\boldsymbol{\beta}}_3\end{aligned} \tag{3.40}$$

This approach faces similar problems as (3.37) but in addition it is also not guaranteed that tasks in the middle, in this case $\dot{\boldsymbol{\beta}}_2$, will not be violated by lower priority tasks, since the null space projectors are not commutative.

A correct solution is to use the *augmented Jacobian* [33], also called the *extended Jacobian* [7]:

$$\boldsymbol{A}_{12} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix}, \quad \boldsymbol{N}_{12} = \boldsymbol{N}(\boldsymbol{A}_{12}) \tag{3.41}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = \boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1 + (\boldsymbol{A}_2\boldsymbol{N}_1)^+\dot{\boldsymbol{\beta}}_2 + (\boldsymbol{A}_3\boldsymbol{N}_{12})^+\dot{\boldsymbol{\beta}}_3 \tag{3.42}$$

Based on the method proposed for two tasks in [70], a formalism for multiple tasks in a general framework was the first time presented in [105] for arbitrary priority levels. This algorithm was further improved in [5], where the nullspace projection matrix was computed recursively to safe computational time, resulting in the following recursive formulation

$$\dot{\boldsymbol{q}}_{v0} = \boldsymbol{0}, \qquad \boldsymbol{N}_0 = \boldsymbol{I}_n \tag{3.43}$$

$$\hat{\boldsymbol{A}}_i = \boldsymbol{A}_i \boldsymbol{N}_{i-1} \tag{3.44}$$

$$\boldsymbol{N}_i = \boldsymbol{N}_{i-1} - \hat{\boldsymbol{A}}_i^{+} \hat{\boldsymbol{A}}_i \tag{3.45}$$

$$\dot{\boldsymbol{q}}_{v_i} = \dot{\boldsymbol{q}}_{v_{i-1}} + \hat{\boldsymbol{A}}_i^{+}(\dot{\boldsymbol{\beta}}_i - \boldsymbol{A}_i \dot{\boldsymbol{q}}_{v_{i-1}}) \tag{3.46}$$

This is the common formula and it is considered the standard approach for realizing multiple hierarchical tasks.

A detailed discussion and comparison of successive and augmented projection as well as stability analysis for velocity controlled systems can be found in [3]. An approach based on the transposed of the Jacobian is covered in [18], which is neglected here due to its significantly poorer performance [5].

3.2.3. Joint Limits

A mechanical joint q_i is usually bounded to a validity domain $q_{i_m} \leq q_i \leq q_{i_M}$. While general inequality constraints or tasks will be treated in more detail in Chapter 4, it is important to mention that joint limits are a special case. Since first, they represent a critical constraint which has to be obeyed rigorously. Second, unlike other inequality tasks, it can already be easily incorporated into the recursive nullspace mapping approach (3.43) – (3.46).

A straight forward approach to maintain joint limits is clamping, which is also usually implemented in low-level to avoid physical damage:

$$q_{i_d} := \begin{cases} q_{i_m}, & \text{if } q_{i_d} < q_{i_m} \\ q_{i_M}, & \text{if } q_{i_d} > q_{i_M} \\ q_{i_d}, & \text{else} \end{cases}, \tag{3.47}$$

where q_{i_M} and q_{i_m} are the lower and upper limits of joint q_i. In [12] it is shown that such simple clamping will result in a non optimal final state and Fig. 3.6 visualizes the occurring problem.

In [80], a trade-off between joint limit avoidance and the motion objective is introduced. In [17], a weighted least-norm solution is used to damp the joint velocity when a joint

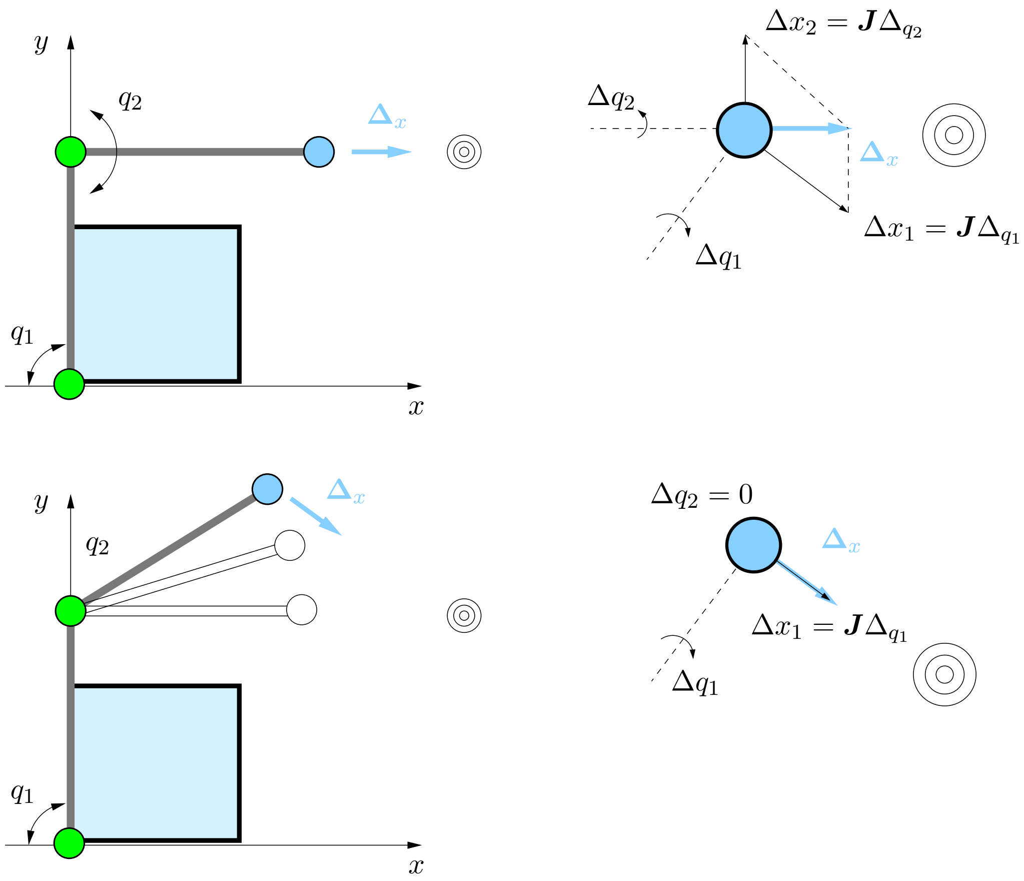


Figure 3.6. The task is to reach a point on the plane. The drawing indicates the directions of the instantaneous effector variations induced by joint variations. The top right corner shows the solution found for the desired variation $\mathbf{\Delta x}$. However, as clamping leads to ignoring the Δq_1 variation, the forearm moves up over the next few iterations until the Δq_2 contribution is reduced to zero, resulting in convergence to a non-optimal state (image concept taken from [6]).

is close to its limit. Both solutions behave improperly when motion objective and the avoidance criteria become incompatible [34].

A simple and effective solution is proposed in [6] within the context of animation trajectory planning. In [6], where joint increments $\boldsymbol{\Delta q}$ are computed instead of joint velocities, it is proposed to check if the new joint position $q_i + \Delta q_i$ exceeds its limits by δq (see Fig. 3.7). If that is the case, the ith column of all the involved task Jacobians are set to $\mathbf{0}$, removing the effect of joint q_j and treating it as a fixed connection. The inverse kinematics computation is performed again until a solution without joint limit violation is found.

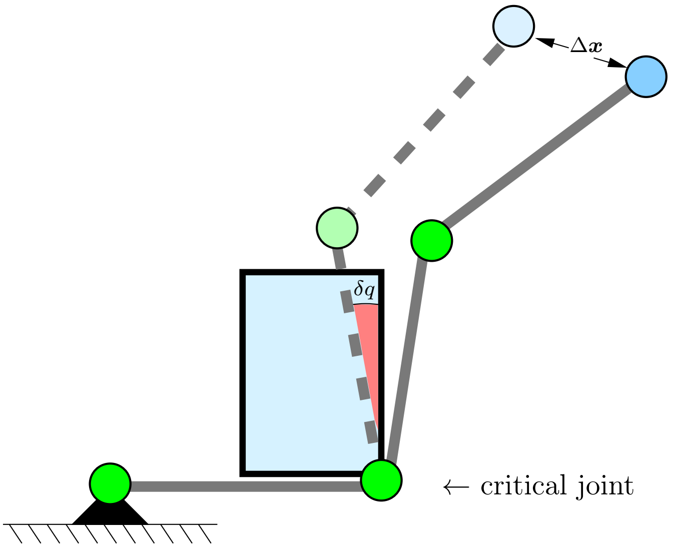


Figure 3.7. Treating joint limits as proposed in [6]. As desired positions are commanded within an animation trajectory planner, it is checked whether the new desired command is violating a joint limit. If this is the case, the respective critical joint is clamped by effectively removing its influence on the final solution.

In order to compute instantaneous velocity commands, the approach from [6] has to be slightly modified. Instead of checking the new joint position, we define a critical area near each joint limit (see Fig. 3.8) and check if the computed joint velocity is driving the joint towards its limit. If this is the case, the joint is clamped by setting the associated column in $\boldsymbol{A}$ to $\mathbf{0}$ and repeat the computations (3.44) – (3.46) until a valid solution is found. This approach is summarized in algorithm C.2 in appendix C. Instead of instantaneous clamping, the impact of a critical joint could be gradually scaled down to avoid discontinuities in the joint velocities, as done in [91].

The presented approach fails in the presence of dynamic joint limits, e.g. due to self collision. The reason for this is, that a critical joint is only clamped without actively avoiding the limit. A solution to this is to apply artificial potential fields [56], which states also an approach for general inequality tasks what will be discussed in depth in Chapter 4.

3.2.4. Algorithmic Singularities

In Chapter 2 kinematic singularities were described. When combining multiple Cartesian tasks in the way it was done in the previous section, algorithmic singularities need to be

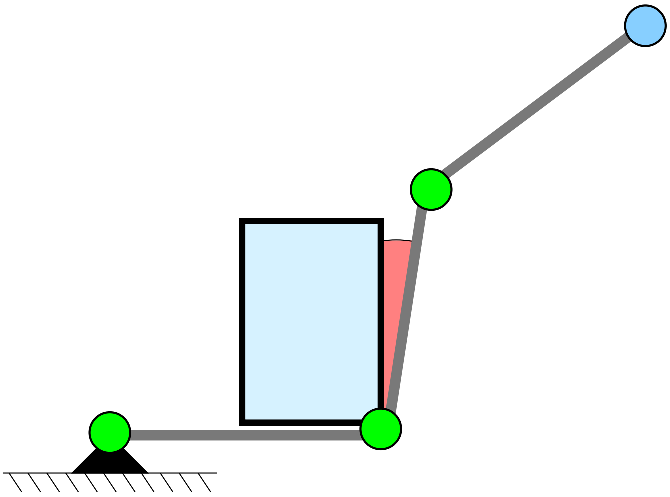


Figure 3.8. For a velocity interface, a critical threshold before a joint limit is defined. If this threshold is crossed and the computed joint velocity is pointing towards the limit, the respective joint is clamped.

taken into account as well. These appear if a lower priority task is not achievable due to the higher priority tasks. As with kinematic singularities, the problematic configurations are the ones close to the singularity, which result in excessive joint velocities (see Fig. 3.9).

Applying the nullspace mapping after the computation of the joint velocities for the second task is robust to algorithmic singularities, what is depicted in Fig. 3.9. In addition, following the joint limit strategy presented in the previous section increases the possibility for an algorithmic singularity, since clamping a joint removes one degree of freedom, which could introduce additional singularities (see Fig. 3.10). As these singularities are not as easy to avoid as kinematic singularities, proper treatment of such situations is crucial.

The same damping techniques as reviewed in Sec. 2.2.2 can be applied to tackle algorithmic singularities. Instead of computing $\boldsymbol{A}^{+}$, the damped inverse $\boldsymbol{A}^{+\lambda}$ is computed as for the kinematic case in (2.17). However, it is important to use the undamped pseudo-inverse for the computation of $\boldsymbol{N}(\boldsymbol{A})$, since the damped version lacks various vital properties [5]. A low-cost computation is

$$\boldsymbol{N}(\boldsymbol{A}) = \boldsymbol{I}_n - \sum_{i=1}^{r} \boldsymbol{v}_i \boldsymbol{v}_i^T, \tag{3.48}$$

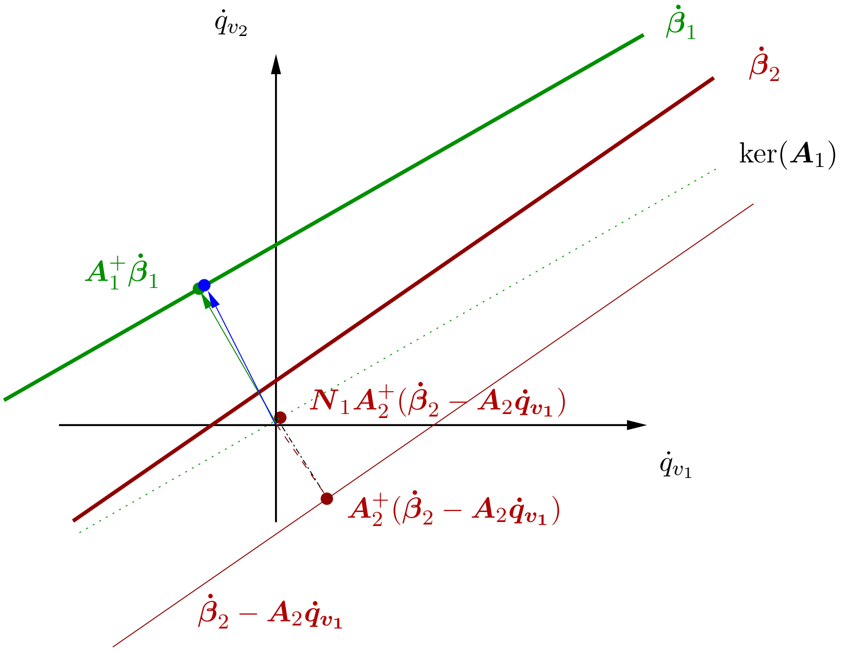

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_1$
$\dot{\boldsymbol{\beta}}_2$
$\ker(\boldsymbol{A}_1)$
$\boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1$
$\boldsymbol{N}_1\boldsymbol{A}_2^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1})$
$\dot{q}_{v_1}$
$\boldsymbol{A}_2^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1})$
$\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1}$

Figure 3.9. Visualization of an algorithmic singularity for the 2 DoF case and a possible work-around. Each of the two tasks is perfectly achievable. However, hierarchical combination via (3.39) results in a configuration close to an algorithmic singularity. As in the proximity of kinematic singularities, a solution to the problem exists (intersection of the red and green line), albeit it requires high joint velocities. Applying the nullspace mapping after the computation of the least norm solution (3.37) (blue arrow) is robust against such algorithmic singularities for the cost of accuracy in the lower priority tasks.

where $\boldsymbol{v}_i$ refers to the ith column of the $\boldsymbol{V}$ matrix in the SVD of $\boldsymbol{A}$.

3.2.5. Task Specification

With the given assumptions and the so far developed formalism, a task can be specified by declaring:

- the task type (or task Jacobian $\boldsymbol{A}$)
- desired task variable $\boldsymbol{\alpha}_d$
- the feedforward differential task variable $\dot{\boldsymbol{\alpha}}_{\text{ff}}$
- the convergence factor $\boldsymbol{\Lambda}$

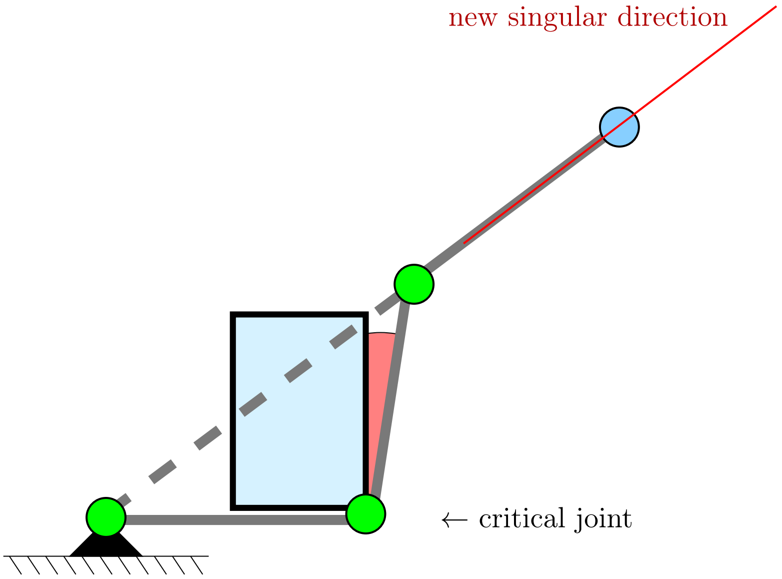


Figure 3.10. Clamping a joint to react on a potential limit violation as proposed in [6] introduces additional singularities.

- the subspace matrix $\boldsymbol{S}$.
- the id of the kinematic chain

The id of the kinematic chain is relevant if more than one target frames are defined. This is the case if multiple end-effectors are regulated in a kinematic tree or if another point on the robot is regulated in Cartesian space, e.g. the elbow joint.

3.2.6. Findings

As the regulation problem is stated in the general linear form (3.21), recursive nullspace mapping approaches can be directly applied, which are commonly used when facing multiple tasks [103, 61, 31].

To achieve robustness to algorithmic singularities, it was decided to use a slightly altered version of the recursive nullspace projection method from [6], i.e. the nullspace map was applied after computing the contribution of the lower priority task while compensating for the effects of the higher priority task. This is a good compromise between task accuracy and robustness. The recursive formulation is

$$\dot{\boldsymbol{q}}_{v_0} = \boldsymbol{0}, \qquad \boldsymbol{N}_0 = \boldsymbol{I}_n \tag{3.49}$$

$$\boldsymbol{N}_i = \boldsymbol{N}_{i-1} - \boldsymbol{A}_i^+ \boldsymbol{A}_i \tag{3.50}$$

$$\dot{\boldsymbol{q}}_{v_i} = \dot{\boldsymbol{q}}_{v_{i-1}} + \boldsymbol{N}_{i-1}\boldsymbol{A}_i^+(\dot{\boldsymbol{\beta}}_i - \boldsymbol{A}_i\dot{\boldsymbol{q}}_{v_{i-1}}) \tag{3.51}$$

and a visualization is given in Fig. 3.11. The resulting algorithm C.1 can be found in appendix C.

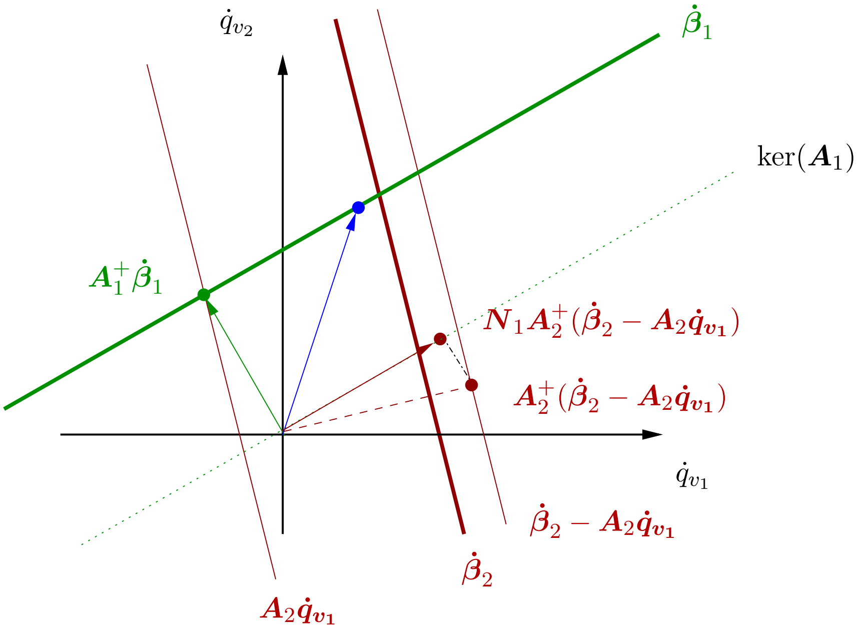
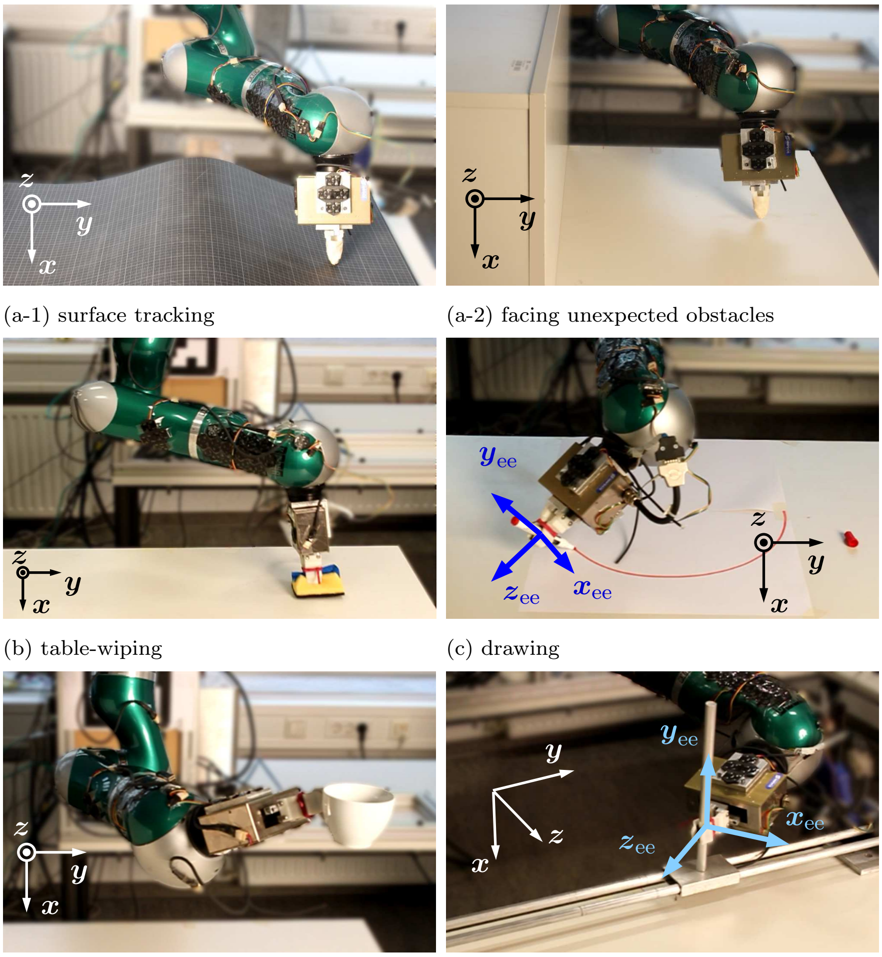

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_1$
$\ker(\boldsymbol{A}_1)$
$\boldsymbol{A}_1^+\dot{\boldsymbol{\beta}}_1$
$\boldsymbol{N}_1\boldsymbol{A}_2^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1})$
$\boldsymbol{A}_2^+(\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1})$
$\dot{q}_{v_1}$
$\dot{\boldsymbol{\beta}}_2 - \boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1}$
$\dot{\boldsymbol{\beta}}_2$
$\boldsymbol{A}_2\dot{\boldsymbol{q}}_{\boldsymbol{v}_1}$

Figure 3.11. Nullspace mapping as used in the present work. The effects of the higher priority tasks are compensated for, but opposed to (3.43) – (3.46), the nullspace mapping is applied after the computation of the least norm solution for the lower priority tasks to achieve robustness to algorithmic singularities.

3.3. Experimental Verification

In this section, two implementations are reported. The first is a proof of concept on a high precision torque controlled manipulator, i.e. the LBR. With this set of experiments it is intended to test the practical applicability of the derived formulation, investigate the tracking error due to the simplified assumptions and to demonstrate the versatility of the resulting task specification interface. The second implementation is conducted on a conceptually completely different robot. A partially compliant dual arm system without



(a-1) surface tracking

(a-2) facing unexpected obstacles

(b) table-wiping

(c) drawing

(d) cup-holding

(e) constrained manipulation

Figure 3.12. Example tasks with mixed positioning and force components

direct torque sensing, i.e. the iCub humanoid robot. This implementation should mainly highlight the general platform independence of the derived formulation.

Details on the robotic hardware can be found in appendix B. For all experiments, $\mathbf{\Lambda}$ was chosen heuristically and is not explicitly stated for the sake of better readability.

3.3.1. Kuka Lightweight Robot

The following examples have been implemented to show how the presented approach can be used to program a variety of tasks by combining positioning- and force-type subtasks in joint and Cartesian space. The experimental setups are depicted in Fig. 3.12.

Table 3.2. Set of subtasks for surface tracking

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$
1	wrench	$8N$	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\begin{bmatrix} y_{\text{init}} + R\cos(2\phi\pi t) - R \\ z_{\text{init}} + R\sin(2\phi\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$\begin{bmatrix} 0\ 0\ 0\ 0\ 0 \\ \boldsymbol{I}_5 \end{bmatrix}$
3a	joint position	$\boldsymbol{0}$	$\boldsymbol{I}_7$
3b	joint torque	$\boldsymbol{0}$	$\boldsymbol{I}_7$

Surface Tracking (Fig. 3.12(a)) This is a classical contact task, where the manipulator is supposed to exert a constant force on a surface while moving along a certain trajectory. Here, a circular trajectory is tracked, starting at $\boldsymbol{p}_{\text{init}}$ with radius R and frequency ϕ in the y-z-plane, while keeping the constant initial orientation $\boldsymbol{o}_{\text{init}}$. The task is summarized in table 3.2. As for the third subtask, one could either choose a positioning task, keeping the joints away from their limits ($\boldsymbol{q}_{v_d} = \boldsymbol{0}$) or alternatively minimizing the joint torques ($\boldsymbol{\tau}_d = \boldsymbol{0}$).

The first trial was conducted on a curved surface with unknown flexibility (see Fig. 3.12(a-1)). The impacts of execution speed and stiffness $\boldsymbol{K}$ of the IFC are demonstrated with this example. The tracking errors of the force and positioning subtasks are plotted in Fig. 3.15 for varying stiffness and in Fig. 3.14 for varying execution speed. As stated in Sec. 3.1.1, the quality of position tracking increases with higher entries in $\boldsymbol{K}$ and decreases for faster execution speeds. With progressive execution speed, one observes the influence of dynamic effects, which are not reflected in $\boldsymbol{\tau}_s$. This is the major downside when using an IFC scheme.

The second experiment was conducted with an unknown obstacle blocking the path of the manipulator (see Fig. 3.12(a-2)). Here the advantage of IFC shows up. Due to its capabilities of handling such unexpected collisions, the manipulator remains stable and gives, for example some high-level application enough time to react to the event. Also, if the joint torque minimization subtask is specified, the nullspace of the higher priority subtasks is used to compensate for collisions occurring at the "elbow"-joint.

Table Wiping (Fig. 3.12(b)) This task is similar to the surface tracking and is an example of a real-world task, where neither very accurate position, nor force tracking is required. The end-effector is supposed to track a sinusoidal trajectory back and forth in

Table 3.3. Set of subtasks for table wiping

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$
1	wrench	$8N$	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\begin{bmatrix} y_{\text{init}} + R_1\cos(2\phi_1\pi t) - R_1 \\ z_{\text{init}} + R_2\sin(2\phi_2\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$\begin{bmatrix} 0\ 0\ 0\ 0\ 0 \\ \boldsymbol{I}_5 \end{bmatrix}$
3	joint torque	$\boldsymbol{0}$	$\boldsymbol{I}_7$

the y-z-plane, determined by the amplitudes R_1 and R_2 with associated frequencies ϕ_1 and ϕ_2. The task description is summarized in table 3.3.

Circle Drawing (Fig. 3.12(c)) This is also a modification of the surface tracking task and demonstrates how simple it is to incorporate high-level knowledge by setting an appropriate subspace matrix $\boldsymbol{S}$. With the pen being aligned with the end-effector y-axis y_{ee}, the positioning subtask is invariant to rotations around y_{ee}. Hence the task is defined equally to the surface tracking example, despite that the rotational part is described in terms of rotations around the x_{ee} and z_{ee} end-effector axes only. This relaxation of the task constraints, gives the lower priority tasks more freedom, resulting in smaller joint velocities and a smoother motion due to the successful avoidance of a joint limit, which is hit if this relaxation is not applied. The resulting discontinuity is depicted in Fig. 3.13. Here, $\boldsymbol{\alpha}_2$ is defined in the global frame and then expressed in the $\boldsymbol{S}_2$-system by premultiplication with $\boldsymbol{S}_2^T$, denoting the transposed of the subspace matrix for the second subtask. The third subtask is to keep the joints away from their limits. Table 3.4 shows the task parameters.

Table 3.4. Set of subtasks for circle drawing

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$
1	wrench	$6N$	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\boldsymbol{S}_2^T \begin{bmatrix} 0 \\ y_{\text{init}} + R\cos(2f\pi t) - R \\ z_{\text{init}} + R\sin(2f\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$\boldsymbol{S}_2 = \text{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee}) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3	joint position	$\boldsymbol{0}$	$\boldsymbol{I}_7$

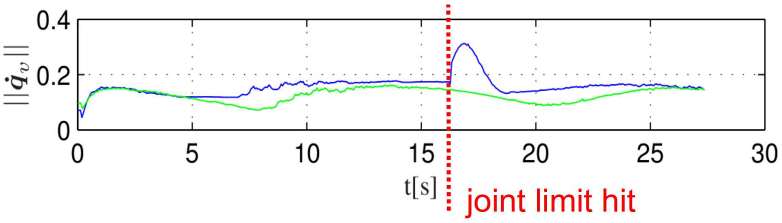


Figure 3.13. Norm of the joint velocities for different task specifications for circle drawing. Blue: keeping constant orientation. Green: Rotation around end-effector y-axis permitted. The additional degree of freedom allows the avoidance of the joint limit, preventing clamping and the resulting discontinuity in the joint velocities.

Cup-Holding (Fig. 3.12(d)) Highest priority is given to a controller holding some fixed orientation. Lower priority tasks can now be defined in any way, e.g. to reach a certain point, react to external sensor information etc. without considering orientation anymore. Choosing for example minimization of joint torques as secondary task, making it possible to push the manipulator around manually. See table 3.5 for the task description.

Table 3.5. Set of subtasks for cup-holding

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$
1	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$
2	joint torque	$\mathbf{0}$	$\boldsymbol{I}_7$

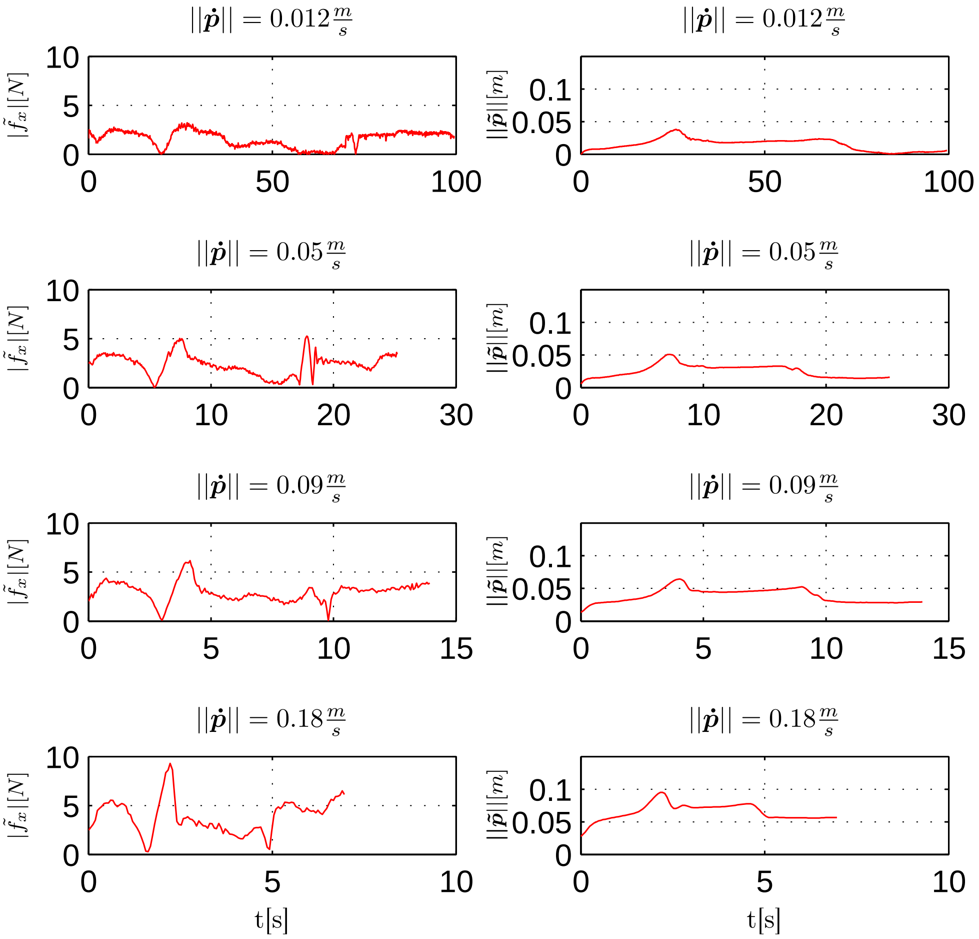


Figure 3.14. Force and positioning error for varying execution speed and $\boldsymbol{K} = 400Nm/\text{rad}$ during a surface tracking task. The quality of position tracking decreases for faster execution speeds. Also, with progressive execution speed, one observes the influence of dynamic effects, which is reflected in increasing force error.

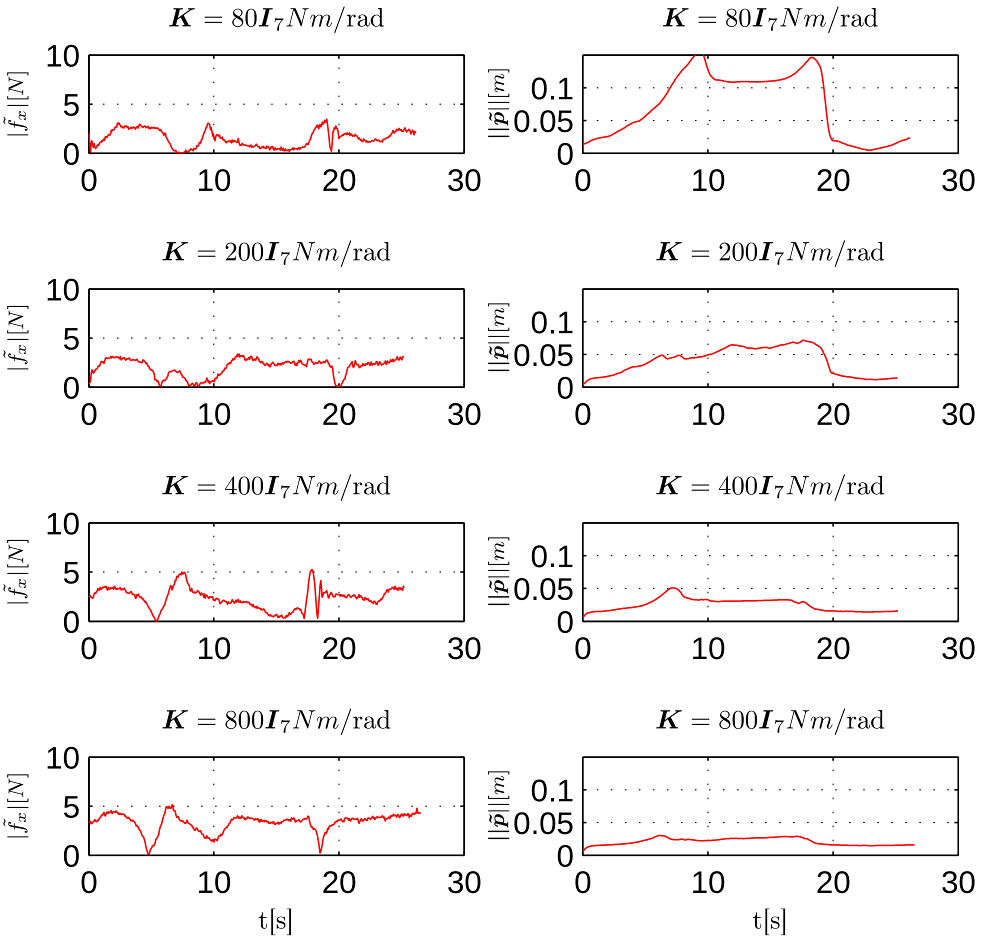


Figure 3.15. Force and positioning error for varying virtual joint stiffness and $||\dot{\boldsymbol{p}}|| = 0.05\frac{m}{s}$ during a surface tracking task. Higher stiffness results in better position tracking in trade for reduced compliance. A slightly higher average force error can be observed for higher stiffness, which is due to an increased sensitivity of $\boldsymbol{h}$ to $(\boldsymbol{q}_v - \boldsymbol{q})$.

3.3.2. iCub

3.3.2.1. Characteristics of the iCub Platform

The iCub is not equipped with torque sensing in every joint and uses the measurements of a 6-axis force/torque sensor in its shoulder joint together with the current in the motors to estimate the joint torques and achieve compliance in some joints. The resulting impedance controller is less accurate than its counterpart on the LBR. In addition, the three wrist joints are not compliant, which prohibits regulation of forces as described in Sec. 3.1 with a contact point at the hands. Nevertheless, the developed task specification scheme allows easily to implement different workarounds to overcome these issues.

The stiffness and damping values were chosen heuristically as $50\frac{Nm}{\text{rad}}$ and $0.5\frac{Nms}{\text{rad}}$ for the hip joints and to $5\frac{Nm}{\text{rad}}$ and $0.2\frac{Nms}{\text{rad}}$ for the others. No additional offset torque was commanded ($\boldsymbol{\tau}_{\text{offset}} = \mathbf{0}$).

To account for the stiff joints in the robotic structure, it is defined that force tasks are executed only with the compliant joints. Therefore, four main kinematic chains are defined, two for each arm, starting at the hip. For every arm, one chain ends at the wrist in a predefined contact point, where force tasks can be executed and one at the palm of the hand, used for pose regulation only (see Fig. B.3).

Another issue arises from the inaccurate impedance rendering and estimation of the external torques in combination with a higher priority force task. With the weight of the whole torso on joints 2 and 3, a major drift is likely to occur while trying to compensate for erroneously estimated external torques in a “bended” posture of the torso. The simplest solution is to clamp these joints via a high priority positioning task, whenever force tasks are specified. It has to be mentioned, that while removing these degrees of freedom avoids the drift issue, it considerably limits the workspace of the robot.

3.3.2.2. Applications

Two contact tasks have been implemented. For the task specification $\bar{\boldsymbol{q}}$ describes a desirable configuration, corresponding to the posture in Fig. B.3 right, where most joints are in the middle of their workspace, the hand is outstretched and the weight of the torso is compensated by the structure instead of the hip joints.

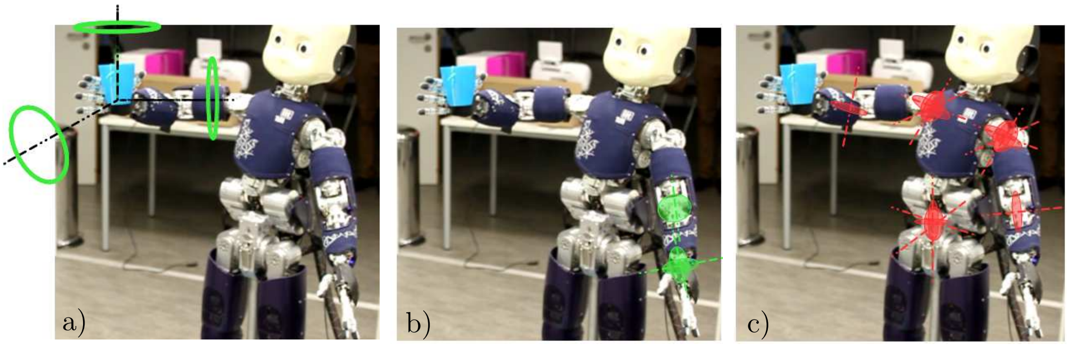


Figure 3.16. Main subtasks for holding a cup while minimizing the static interaction torques, i.e. holding a constant orientation (a), holding some constant position with the non-compliant joints (b) and minimizing the joint torques (c).

Compensating External Torques while Holding a Cup The first task is to hold the orientation of a cup constant, while minimizing the static interaction torques. The main task is composed of three subtasks. The first is to keep the initial orientation of the right hand, holding the cup. The second subtask is to keep the non-compliant wrist joints away from their limits. The third subtask is to minimize the static joint torques. The second subtask is necessary, since torque tasks are not defined for the non-compliant joints. The task specification is summarized in table 3.6 and depicted in Fig. 3.16.

Table 3.6. Set of subtasks for cup-holding

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$	chain ID
1	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$	RH
2	joint position	$\begin{bmatrix} \bar{q}_8 \\ \bar{q}_9 \\ \bar{q}_{10} \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{0}_{7\times3} \\ \boldsymbol{I}_3 \\ \boldsymbol{0}_{7\times3} \end{bmatrix}$	NA
3	joint torque	$\boldsymbol{0}$	$\boldsymbol{I}_{17}$	NA

Wiping a Table The second implementation is keeping the full pose of the right hand constant, but this time the other arm should be used to wipe a table. The task specification can be found in table 3.7 and is depicted in Fig. 3.17. The first subtask is to overcome the drift-issue as described in the previous section. Tasks 2–4 are the table-wipe subtasks. The orientation and x-position of the left arm should remain constant while the desired y-position is determined by an offset value, which switches after a certain amount of time, so that the arm alternates between two points, while applying a contact force in z-direction. The right hand should maintain its initial pose, where the orientation has a higher priority than position (tasks 5 and 6). The final task is to keep the joints in the convenient configuration $\bar{\boldsymbol{q}}$.

Table 3.7. Set of subtasks for table-wiping and cup-holding

task	type	$\boldsymbol{\alpha}_d$	$\boldsymbol{S}$	chain ID
1	joint position	$\begin{bmatrix} \bar{q}_2 \\ \bar{q}_3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ \multicolumn{2}{c}{\mathbf{0}_{14\times 2}} \end{bmatrix}$	NA
2	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$	LH
3	cart. pose	$\begin{bmatrix} x_{\text{init}} \\ y_{\text{init}} + y_{\text{offset}}(t) \end{bmatrix}$	$\begin{bmatrix} \boldsymbol{I}_2 \\ \mathbf{0}_{4\times 2} \end{bmatrix}$	LH
4	wrench	$-7N$	$[0\ 0\ 1\ 0\ 0\ 0]^T$	LW
5	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$	RH
6	cart. pose	$\boldsymbol{p}_{\text{init}}$	$\begin{bmatrix} \boldsymbol{I}_3 \\ \mathbf{0}_3 \end{bmatrix}$	RH
7	joint position	$\bar{\boldsymbol{q}}$	$\boldsymbol{I}_{17}$	NA

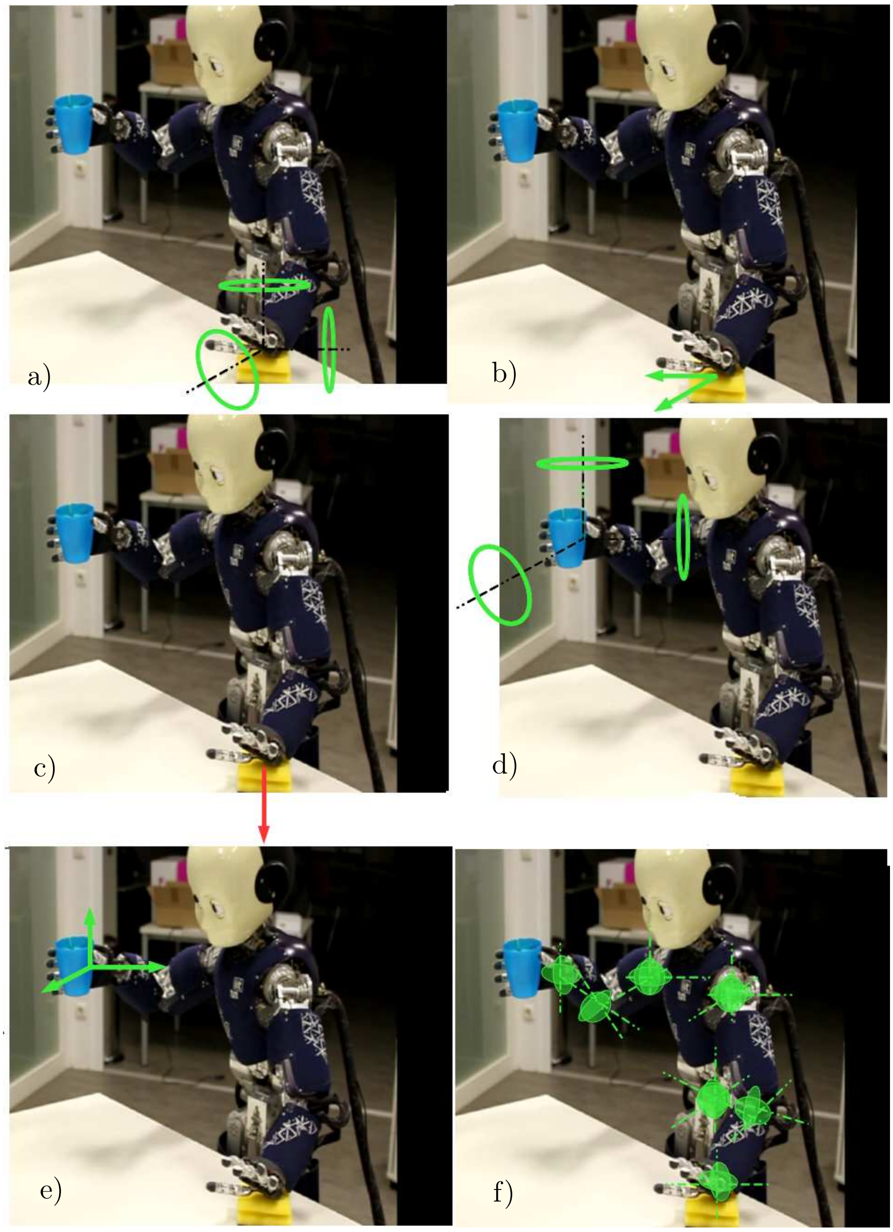


Figure 3.17. Main subtasks for holding a cup with one hand, while wiping a table with the other. For the hand wiping the table, the following tasks are defined: maintaining a constant orientation (a), time triggered point-to-point motion (b) and applying a contact force (c). The subtasks for the other hand are keeping a constant orientation (d) and position (e). Finally, the robot should stay as close as possible to the convenient configuration $\bar{\boldsymbol{q}}$ (f).

3.4. Concluding Remarks

In this chapter the problem of simultaneous force and positioning regulation for IFC controlled robots is approached and the resulting task specification formalism for hierarchical force and positioning tasks on joint- and Cartesian-level is presented. For this purpose, task variables with associated task Jacobians in the context of IFC are defined and a generalized task regulation scheme is proposed. Then, recursive nullspace mapping is applied to combine multiple tasks in a hierarchical way. The underlying simplifying assumptions trade accuracy for generalization of force and position regulation while keeping the general compliance of the indirect force controlled robot. The practical applicability and general platform independence was investigated experimentally.

Potential pitfalls of the proposed approach are:

- Wrong computation of the contact wrench near singular configurations due to the resulting high sensitivity of $\boldsymbol{h}$ to $(\boldsymbol{q}_v - \boldsymbol{q})$.
- Wrong estimation of contact forces if a contact does not occur at the end of the kinematic chain.
- Bad force tracking for high joint velocities as dynamic effects are not taken into account.

However, it is believed that these issues are tolerable and still leave a large range of potential applications. A major drawback however, is that with the formulation presented so far, only equality tasks can be specified. Therefore the next step is to extend the scheme to support inequality tasks, what is the main focus of the next chapter.

Chapter 4

Unified Equality and Inequality Task Specification and Stable Set-Point Generation

Inequality constraints or inequality tasks are of particular interest for robot programming, since they capture inherent limits of the robot like mechanical joint, velocity, acceleration and torque limits. Also, it is more suitable to define tasks as inequalities, e.g. to prevent the robot from entering certain areas in the workspace or adding a certain tolerance to a task variable instead of a specific desired value. In a multiple tasks framework, this second point is especially useful, as it relaxes the constraints on a task and gives more freedom to lower priority ones.

The major goal of this chapter is to improve the formulation, derived in the previous chapter, by integrating inequality tasks and ensure stability of the overall system. In Sec. 4.1 it is first shown how the robot regulation can be formulated as an optimization problem and an approach based on state of the art techniques is presented to extend the task specification scheme from Chapter 3 to support equality and inequality tasks in a unified way. In Sec. 4.2 the stability of the overall system is analyzed from the perspective of passivity theory and it is shown how to enforce stability by modulating the set-points generated by the SPG. Sec. 4.3 validates the derived formalism with different conceptual applications on the LBR and Sec. 4.4 concludes the chapter.

4.1. Hierarchical Inequality Tasks

4.1.1. State of the Art

Due to computational limitations, only equality tasks are considered in the classical redundancy resolution approaches. However, there exist a vast number of resolving joint-level inequality constraints (see Sec. 3.2.3 for joint position limits). In [4] joint velocity limits are treated in particular.

In the past, specific inequality constraints, like collision/singularity avoidance, have been treated via the gradient projection method [64]. Another popular approach to avoid collisions with external obstacles as well as self collisions is the artificial potential field method introduced in [55]. Inequality position constraints are transformed to equality constraints. Therefore this approach is in general not appropriate for top-level constraints in general. To implement inequality tasks in a higher priority-level, several methods have been proposed. In [102] repulsive fields are added to the highest priority task-level with the drawback of suboptimal solutions due to the already mentioned conversion of inequality tasks to equality tasks. The approach in [84] is similar to [102] and [55]. It combines the task-priority framework with repulsive fields to avoid obstacles for reaching motions.

To obey general inequality constraints it was proposed to integrate them into a linear program, which encodes the task. This was the fundamental idea in [28, 111, 81]. However, no hierarchy was imposed among the different inequality tasks, resulting in problems if the tasks are contradicting (see also Sec. 3.2 for a brief discussion).

Recently, Flacco et. al. introduced an algorithm to incorporate joint angle, velocity and acceleration limits and exploit them as good as possible to achieve a Cartesian task by scaling it appropriately [41]. A unified, but computationally expensive approach is presented in [72], where general inequality tasks are treated on every priority-level in a stack-of-tasks framework. The first approach to solve hierarchical optimization problems as a cascade of single-objective optimization problems was proposed in [9]. The general idea was applied to quadratic problems in [51] and [35], which is a generalization of the *quadratic programming* (QP) approach in [36]. All these schemes are defined on the kinematic-level only and so far, there is no application in the context of IFC.

Due to its generic formulation, the approach proposed in [51] was chosen as the basis for the solution proposed in this thesis. The main advantage of the QP approach is, that it provides a simple and general formalization which captures joint and task space inequalities in a unified way and can be easily adapted for the case of IFC. The potential violation of the specified inequalities attached to this approach as noted in [41], is not critical in the present case, since comparatively relaxed requirements on accuracy are assumed and also the inherent compliance of the IFC should maintain a certain level of safety. The core idea is to express the stack-of-tasks as a sequential QP, where every problem is solved as good as possible without altering the quality of the solution for the previous tasks.

4.1.2. Instantaneous Robot Control as an Optimization Problem

4.1.2.1. Equality Tasks

An equality task is defined by the linear constraint

$$\dot{\boldsymbol{\beta}}_d = \boldsymbol{A}\dot{\boldsymbol{q}}_v, \tag{4.1}$$

where $\dot{\boldsymbol{\beta}}_d$ and $\boldsymbol{A}$ are given and $\dot{\boldsymbol{q}}_v$ has to be determined. This objective can be formulated as a constrained optimization problem, which obligation is to find a $\dot{\boldsymbol{q}}_v$, obeying the equality constraint (4.1) and minimizing a certain cost-function, e.g. the L_2-norm of $\dot{\boldsymbol{q}}_v$. A possible problem statement is

$$\text{min.} \quad f(\dot{\boldsymbol{q}}_v) = \frac{1}{2}||\dot{\boldsymbol{q}}_v||_2^2 = \frac{1}{2}\dot{\boldsymbol{q}}_v^T\dot{\boldsymbol{q}}_v \tag{4.2}$$

$$\text{s.t.} \quad \boldsymbol{r}(\dot{\boldsymbol{q}}_v) = \boldsymbol{A}\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{\beta}}_d = \boldsymbol{0}, \tag{4.3}$$

where $f(\dot{\boldsymbol{q}}_v)$ is a cost term, which has to be minimized and $\boldsymbol{r}(\dot{\boldsymbol{q}}_v)$ is the equality constraint. This problem can be solved analytically by deriving the Lagrangian as

$$\mathcal{L}(\dot{\boldsymbol{q}}_v, \boldsymbol{\nu}) = f + \boldsymbol{\nu}^T\boldsymbol{r} = \frac{1}{2}\dot{\boldsymbol{q}}_v^T\dot{\boldsymbol{q}}_v + \boldsymbol{\nu}^T(\boldsymbol{A}\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{\beta}}_d), \tag{4.4}$$

with $\boldsymbol{\nu}$ as the Lagrange multiplier or dual optimization variable. As (4.2) is a strictly convex function, applying the *Karush-Kuhn-Tucker* (KKT) conditions [13] results in

$$\nabla\mathcal{L} = \dot{\boldsymbol{q}}_v^* + \boldsymbol{A}^T\boldsymbol{\nu}^* = \boldsymbol{0} \tag{4.5}$$

$$\boldsymbol{A}\dot{\boldsymbol{q}}_v^* = \dot{\boldsymbol{\beta}}_d, \tag{4.6}$$

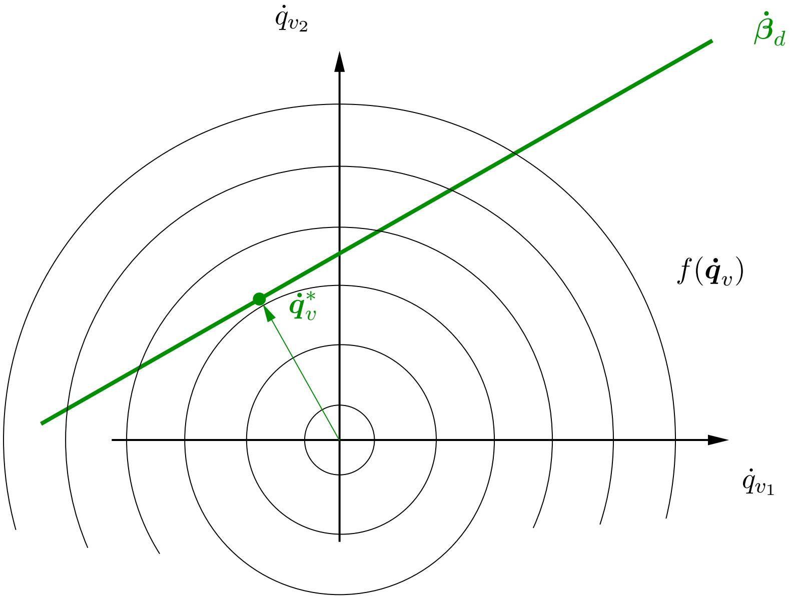

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_d$
$f(\dot{\boldsymbol{q}}_v)$
$\dot{\boldsymbol{q}}_v^*$
$\dot{q}_{v_1}$

FIGURE 4.1. The minimum-norm solution obtained by solving a constrained optimization problem.

where $\dot{\boldsymbol{q}}_v^*$ and $\boldsymbol{\nu}^*$ are the vectors of the optimal primal and dual variables [13]. Provided that rank($\boldsymbol{A}$) $\geq m$ or in other words, there exists a $\dot{\boldsymbol{q}}_v$ to obey (4.3), $\boldsymbol{A}\boldsymbol{A}^T$ is invertible. Combining (4.5) and (4.6) gives

$$\dot{\boldsymbol{q}}_v^* = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\dot{\boldsymbol{\beta}}. \tag{4.7}$$

This solution corresponds to the minimum-norm solution obtained by the Moore-Penrose pseudo-inverse for a fat $\boldsymbol{A}$:

$$\dot{\boldsymbol{q}}_v^* = \boldsymbol{A}^+\dot{\boldsymbol{\beta}}_d \tag{4.8}$$

Fig. 4.1 depicts this equivalency.

If the task, encoded by the equality constraint (4.3) is not achievable, respectively rank($\boldsymbol{A}$) $< m$, the problem is over-constrained. A solution can be found by solving the unconstrained least squares approximation problem

$$\text{min.} \quad f(\dot{\boldsymbol{q}}_v) = ||\boldsymbol{A}\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{\beta}}||_2^2 = \dot{\boldsymbol{q}}_v^T\boldsymbol{A}^T\boldsymbol{A}\dot{\boldsymbol{q}}_v - 2\dot{\boldsymbol{\beta}}^T\boldsymbol{A}\dot{\boldsymbol{q}}_v + \dot{\boldsymbol{\beta}}^T\dot{\boldsymbol{\beta}}, \tag{4.9}$$

which can be solved analytically again, by applying the optimality condition

$$\nabla f = 2\boldsymbol{A}^T \boldsymbol{A} \dot{\boldsymbol{q}}_v^* - 2\boldsymbol{A}^T \dot{\boldsymbol{\beta}} = \boldsymbol{0} \tag{4.10}$$

and solving for $\dot{\boldsymbol{q}}_v^*$, which gives

$$\dot{\boldsymbol{q}}_v^* = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \dot{\boldsymbol{\beta}}. \tag{4.11}$$

This corresponds to the solution obtained by the Moore Penrose pseudo-inverse (4.8).

4.1.2.2. Inequality Tasks

The appealing advantage of formalizing the regulation task as an optimization problem is that it allows to introduce inequality constraints (inequality tasks) to solve a problem like

$$\text{min.} \quad \frac{1}{2}||\dot{\boldsymbol{q}}_v||_2^2 \tag{4.12}$$

$$\text{s.t.} \quad \dot{\boldsymbol{\beta}}_m \leq \boldsymbol{A}\dot{\boldsymbol{q}}_v \leq \dot{\boldsymbol{\beta}}_M. \tag{4.13}$$

Instead of having a particular desired value $\dot{\boldsymbol{\beta}}_d$, the task variable can be specified to stay in a certain desired range $\dot{\boldsymbol{\beta}}_m \leq \dot{\boldsymbol{\beta}} \leq \dot{\boldsymbol{\beta}}_M$. The optimization problem is depicted in Fig. 4.2. There is no analytical way of solving such a problem. However, very fast and robust numerical methods have been developed, which can find a solution to (4.12)–(4.13) in finite time and for which powerful software libraries are already implemented, e.g. [39].

An advantage of relaxing a task constraint by specifying a desired range instead of a particular value is depicted in Fig. 4.3. Given two subtasks with task boundaries $\dot{\boldsymbol{\beta}}_{1_m}$ and $\dot{\boldsymbol{\beta}}_{1_M}$ for the task with highest priority and $\dot{\boldsymbol{\beta}}_{2_m}$ and $\dot{\boldsymbol{\beta}}_{2_M}$ for the task with lower priority. The associated task Jacobians are denoted as $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. The two equality tasks are ill-conditioned. If one of the tasks is redefined as an inequality, a descent solution is found.

In a set of k subtasks, every task is defined as an inequality constraint, where with $\dot{\boldsymbol{\beta}}_d = \dot{\boldsymbol{\beta}}_m = \dot{\boldsymbol{\beta}}_M$ equality constraints can be implemented within (4.13)[1]. All tasks can

[1] State of the art solvers recognize this as an equality constraint and take proper measures.

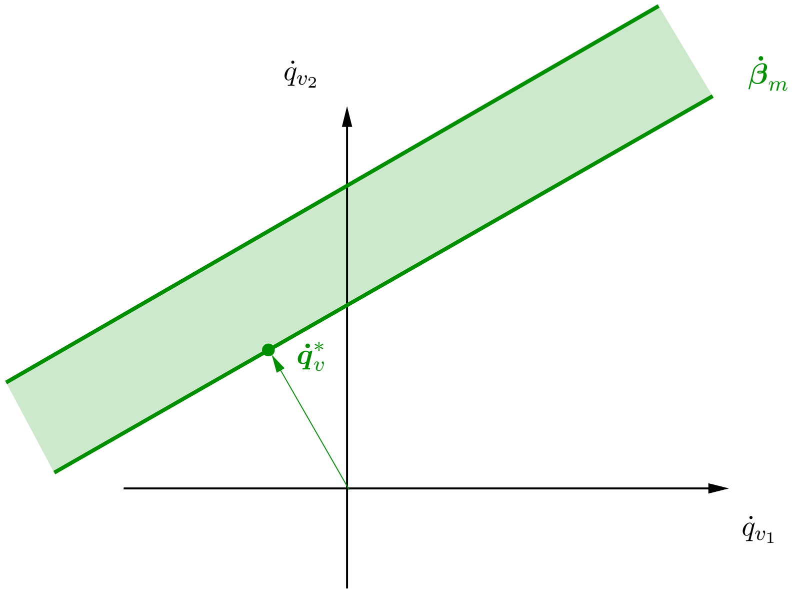


FIGURE 4.2. Formulating a task as an inequality constraint. The admissible set of possible solutions, i.e. to realize a secondary task, is increased if compared to 4.1.

be summarized via

$$\text{min.} \quad \frac{1}{2}||\dot{\boldsymbol{q}}_v||_2^2 \tag{4.14}$$

$$\text{s.t.} \quad \bar{\dot{\boldsymbol{\beta}}}_m \leq \bar{\boldsymbol{A}}\dot{\boldsymbol{q}}_v \leq \bar{\dot{\boldsymbol{\beta}}}_M \tag{4.15}$$

where $\bar{\boldsymbol{A}} = [\boldsymbol{A}_0 \dots \boldsymbol{A}_k]^T$ and $\bar{\dot{\boldsymbol{\beta}}} = [\dot{\boldsymbol{\beta}}_0 \dots \dot{\boldsymbol{\beta}}_k]^T$ are the augmented task Jacobians and boundaries for the k tasks. These constraints define the admissible set for the optimization problem and the numerical solver determines a solution within this set, which minimizes the cost-function. Fig. 4.4 visualizes this.

4.1.2.3. Task Hierarchies

Formulating the problem for multiple tasks as (4.14) – (4.15) shows a good performance if all tasks are feasible. If one or more tasks are not feasible with respect to the others or in other words, the problem is over-constrained, the tasks have to be executed following an order of priority or a weighting strategy has to be applied, trading off the task accuracy of the different subtasks against each other. The prioritization strategy is pursued in thesis (see Sec. 3.2 for a discussion on this). An example can be seen in Fig. 4.5, where two higher priority inequality constraints make a lower priority task infeasible. In such

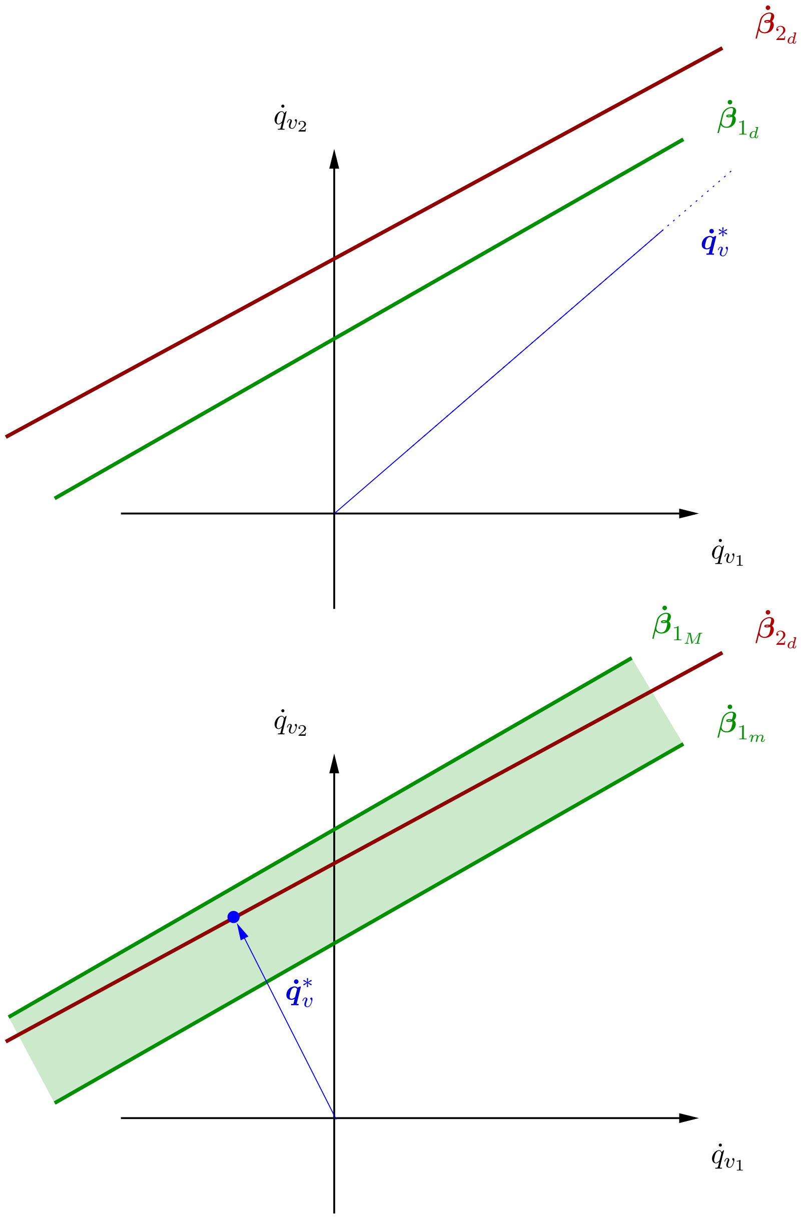

$\dot{\boldsymbol{\beta}}_{2_d}$
$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_{1_d}$
$\dot{\boldsymbol{q}}_v^*$
$\dot{q}_{v_1}$
$\dot{\boldsymbol{\beta}}_{1_M}$
$\dot{\boldsymbol{\beta}}_{2_d}$
$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_{1_m}$
$\dot{\boldsymbol{q}}_v^*$
$\dot{q}_{v_1}$

Figure 4.3. The upper figure shows a classic configuration close to an algorithmic singularity for two equality tasks. If one of the task constraints is relaxed by specifying it as an inequality task (lower figure), a valid solution can be found with descent joint velocities.

a case, the desired solution would be the one that obeys the higher priority tasks and is as close as possible to the new low priority task. Kanoun et. al. [51] proposed to achieve this by solving an optimization problem (i.e. QP) for every subtask, respectively every priority-level separately. For each subtask the inequality constraint is relaxed by reformulating the problem via the usage of *slack variables* [13]. The following example with two inequality tasks should demonstrate the general approach.

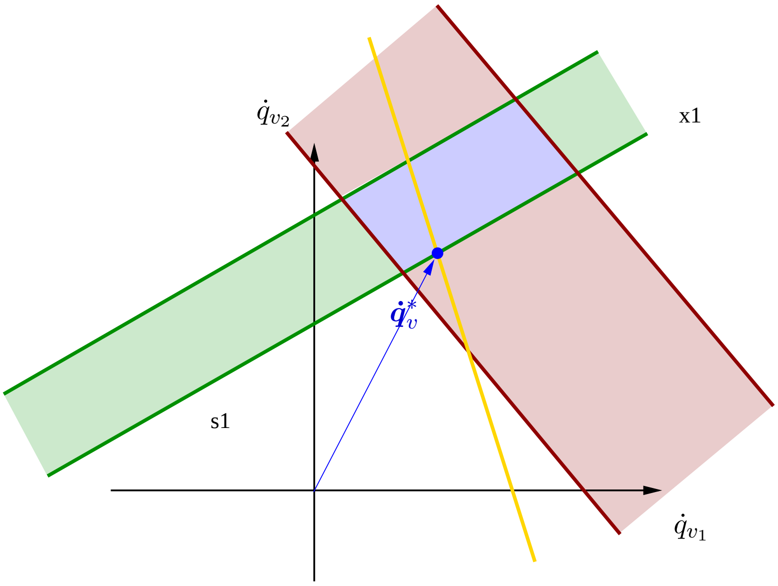


FIGURE 4.4. Every task is formulated as an equality or inequality constraint for the optimization problem.

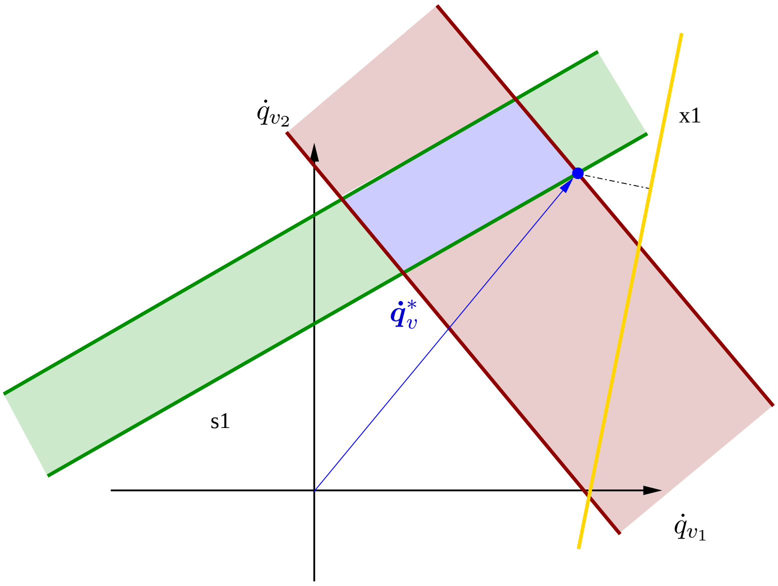


FIGURE 4.5. An unfeasible task is fulfilled as good as possible without violating the higher priority objectives.

For task 1 the modified optimization problem is stated as

$$\text{min.} \quad \frac{1}{2}||\boldsymbol{s}||_2^2 + \frac{1}{2}\rho||\dot{\boldsymbol{q}}_v||_2^2 \tag{4.16}$$

$$\text{s.t.} \quad \dot{\boldsymbol{\beta}}_{1_m} \leq \boldsymbol{A}_1\dot{\boldsymbol{q}}_v - \boldsymbol{s} \leq \dot{\boldsymbol{\beta}}_{1_M}, \tag{4.17}$$

where $\boldsymbol{s}$ denotes a vector of slack variables, which in fact softens the constraint by allowing, yet trying to minimize, a task error represented by $\boldsymbol{s}$. In addition, the joint velocity $\dot{\boldsymbol{q}}_v$ still remains in the cost function as a regularization term with tunable factor ρ, which is important for the numerical stability of the process [77] and is similar to the damped inverse techniques described in Chapter 3. Fig. 4.6 depicts the effect of ρ for the described problem. By making ρ also depend on the lowest singular value of $\boldsymbol{A}$, high joint velocities due to the proximity to a singularity can be tackled as well. This formulation is actually a constrained bi-criterion problem, where ρ tunes the solution on the set of *Pareto-optimal* values [13]. If the subtask is feasible, the optimal vector of slack variables is $\boldsymbol{s}_1^* = \boldsymbol{0}$ and a similar solution $\dot{\boldsymbol{q}}_{v_1}^*$ as for (4.12)–(4.13) is obtained[2], otherwise $\boldsymbol{s}_1^*$ represents the vector of residuals or tolerated errors. Now when formulating the optimization problem for the second task, the optimal solution of the higher priority

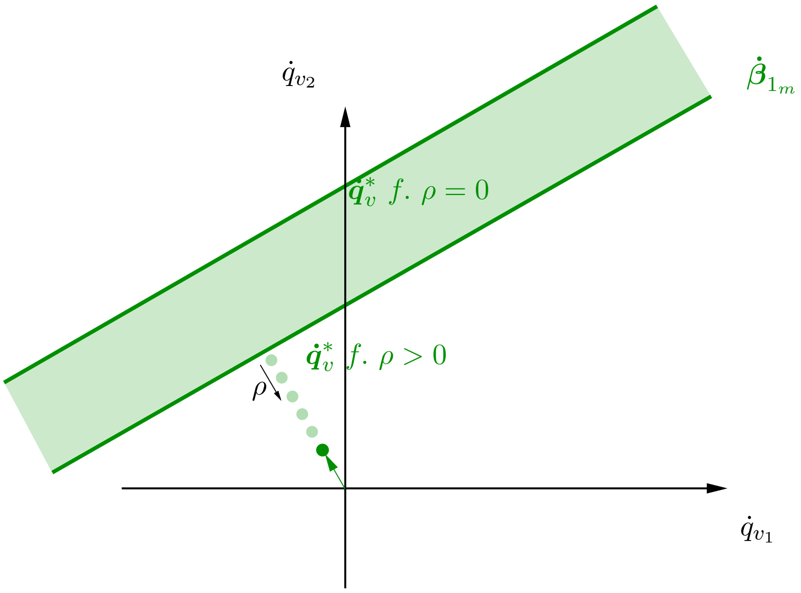

$\dot{q}_{v_2}$
$\dot{\boldsymbol{\beta}}_{1_m}$
$\dot{\boldsymbol{q}}_v^*$ f. $\rho = 0$
$\dot{\boldsymbol{q}}_v^*$ f. $\rho > 0$
ρ
$\dot{q}_{v_1}$

Figure 4.6. If the slack variable formulation is applied, the regularization parameter ρ is important for numerical stability. For $\rho = 0$ every point in the area between the limits would be an evenly good solution.

[2] The regularization term results in a deformation of the task, which is however negligible for small values of ρ

task has to be incorporated as an additional constraint.

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}||\boldsymbol{s}||_2^2 + \frac{1}{2}\rho||\dot{\boldsymbol{q}}_v||_2^2 && (4.18) \\ \text{s.t.} \quad & \dot{\boldsymbol{\beta}}_{2_m} \leq \boldsymbol{A}_2\dot{\boldsymbol{q}}_v - \boldsymbol{s} \leq \dot{\boldsymbol{\beta}}_{2_M} && (4.19) \\ & \dot{\boldsymbol{\beta}}_{1_m} + \boldsymbol{s}_1^* \leq \boldsymbol{A}_1\dot{\boldsymbol{q}}_v \leq \dot{\boldsymbol{\beta}}_{1_M} + \boldsymbol{s}_1^*, . && (4.20) \end{aligned}$$

Note that (4.20) is a hard constraint, not softened up by slack variables. This effectively shifts the task bounds for task 2 by $\boldsymbol{s}_1^*$ and guarantees that the quality of the higher priority task is not declined.

A recursive formulation can be done for $i = [1 \dots k]$ and with the initial values

$$\begin{aligned} \bar{\boldsymbol{A}}_0 &= \boldsymbol{0}_{1\times n} \\ \bar{\dot{\boldsymbol{\beta}}}_{m_0} &= 0 \\ \bar{\dot{\boldsymbol{\beta}}}_{M_0} &= 0 \\ \bar{\boldsymbol{s}}_0^* &= 0. \end{aligned}$$

For every priority-level the following problem has to be solved:

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}||\boldsymbol{s}||_2^2 + \frac{1}{2}\rho||\dot{\boldsymbol{q}}_v||_2^2 && (4.21) \\ \text{s.t.} \quad & \dot{\boldsymbol{\beta}}_{i_m} \leq \boldsymbol{A}_i\dot{\boldsymbol{q}}_v - \boldsymbol{s} \leq \dot{\boldsymbol{\beta}}_{i_M} && (4.22) \\ & \bar{\dot{\boldsymbol{\beta}}}_{i-1_m} + \bar{\boldsymbol{s}}_{i-1}^* \leq \bar{\boldsymbol{A}}_{i-1}\dot{\boldsymbol{q}}_v \leq \bar{\dot{\boldsymbol{\beta}}}_{i-1_M} + \bar{\boldsymbol{s}}_{i-1}^*, && (4.23) \end{aligned}$$

where $\bar{\bullet}_{i-1}$ is the augmented vector/matrix of the previous 0 to $i-1$ values.

4.1.3. General Inequality Tasks

In the previous section, only limits on the compensated differential task variable $\dot{\boldsymbol{\beta}}$ were treated. In the following, it will be shown how these differential limits can be used to implement global limits on the task variable $\boldsymbol{\alpha}$ directly and also how to treat the special case of joint-level limits.

4.1.3.1. Inequality Tasks via Velocity Limits

Instead of having one desired value for the task variable $\boldsymbol{\alpha}_d$, lower and upper bounds ($\boldsymbol{\alpha}_m$ and $\boldsymbol{\alpha}_M$) are specified as a desired range for $\boldsymbol{\alpha}$. This can be defined as an inequality task constraint

$$\boldsymbol{\alpha}_m \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_M. \tag{4.24}$$

Again, an equality task $\boldsymbol{\alpha}_d$ can be specified by setting $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}_M = \boldsymbol{\alpha}_d$.

To keep $\boldsymbol{\alpha}$ inside the range $[\boldsymbol{\alpha}_m,\ \boldsymbol{\alpha}_M]$, a conceptually similar method as in [36] is applied, where the velocity towards an obstacle is damped when the distance reaches a certain threshold. Lower and upper bounds on $\dot{\boldsymbol{\alpha}}$ are induced, described by

$$\dot{\boldsymbol{\alpha}}_m = \boldsymbol{\Lambda}(\boldsymbol{\alpha}_m - \boldsymbol{\alpha}) + \dot{\boldsymbol{\alpha}}_{m_{\text{ff}}} \tag{4.25}$$

$$\dot{\boldsymbol{\alpha}}_M = \boldsymbol{\Lambda}(\boldsymbol{\alpha}_M - \boldsymbol{\alpha}) + \dot{\boldsymbol{\alpha}}_{M_{\text{ff}}} \tag{4.26}$$

depending on the distance of $\boldsymbol{\alpha}$ to $\boldsymbol{\alpha}_m$ and $\boldsymbol{\alpha}_M$ and the convergence factor $\boldsymbol{\Lambda}$. As in (3.27), a feedforward term $\dot{\boldsymbol{\alpha}}_{m_{\text{ff}}}$, respectively $\dot{\boldsymbol{\alpha}}_{M_{\text{ff}}}$ can be optionally specified for both bounds. Fig. 4.7 depicts this approach. When inside the specified range, the differential task variable is bounded by (4.26) and (4.25), so that $\boldsymbol{\alpha}$ stays within the limits. When outside the specified range, $\boldsymbol{\alpha}$ converges towards the nearest limit.

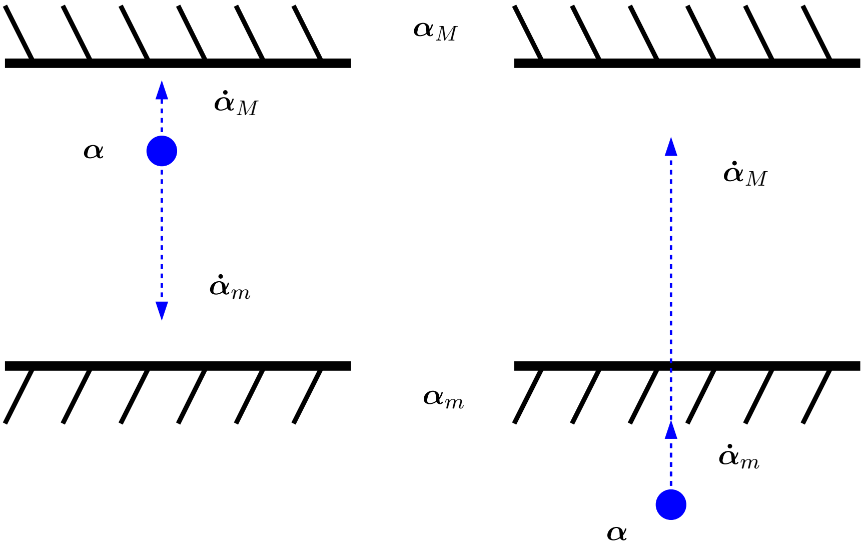

$\boldsymbol{\alpha}_M$
$\dot{\boldsymbol{\alpha}}_M$
$\boldsymbol{\alpha}$
$\dot{\boldsymbol{\alpha}}_m$
$\dot{\boldsymbol{\alpha}}_M$
$\boldsymbol{\alpha}_m$
$\dot{\boldsymbol{\alpha}}_m$
$\boldsymbol{\alpha}$

Figure 4.7. To keep the task variable $\boldsymbol{\alpha}$ inside its bounds $[\boldsymbol{\alpha}_m,\ \boldsymbol{\alpha}_M]$, the velocity $\dot{\boldsymbol{\alpha}}$ is limited to the range $[\dot{\boldsymbol{\alpha}}_m,\ \dot{\boldsymbol{\alpha}}_M]$.

From (4.26) and (4.25), the associated limits on $\dot{\boldsymbol{\beta}}$ can be computed via

$$\dot{\boldsymbol{\beta}}_m = \dot{\boldsymbol{\alpha}}_m + \boldsymbol{\gamma}(\dot{\boldsymbol{q}}) \tag{4.27}$$

$$\dot{\boldsymbol{\beta}}_M = \dot{\boldsymbol{\alpha}}_M + \boldsymbol{\gamma}(\dot{\boldsymbol{q}}). \tag{4.28}$$

The main advantage compared to other common approaches, like artificial potential fields [55], is that a clear inequality constraint is specified, hence no switching or priority shifting is required. Also the limit can be reached in finite time, hence the full range of possible motions is exploited.

4.1.3.2. Special Case: Joint-Level Limits

In addition to the joint position limits treated already in Chapter 3, the joints of every physical manipulator are usually restricted to velocity and acceleration constraints. This can be expressed as the set of inequalities

$$\boldsymbol{q}_m \leq \boldsymbol{q}_v \leq \boldsymbol{q}_M \tag{4.29}$$

$$-\boldsymbol{v}_M \leq \dot{\boldsymbol{q}}_v \leq \boldsymbol{v}_M \tag{4.30}$$

$$-\boldsymbol{a}_M \leq \ddot{\boldsymbol{q}}_v \leq \boldsymbol{a}_M \tag{4.31}$$

where the velocity and acceleration limits are usually symmetric. In general these limits could be captured with the problem statement (4.21)–(4.23) by implementing them as the tasks with the highest priority. However, these limits usually state hard constraints, which should never be exceeded, hence they should not be relaxed using slack variables.

A conventional approach to capture position,velocity and acceleration limits is to use finite differences and shaping the velocity limits according to the most critical value.

$$\dot{\boldsymbol{q}}_{v_m} = \max\{\tfrac{\boldsymbol{q}_m-\boldsymbol{q}}{\Delta T},\ -\boldsymbol{v}_M,\ -\sqrt{2\boldsymbol{a}_M(\boldsymbol{q}-\boldsymbol{q}_m)}\} \tag{4.32}$$

$$\dot{\boldsymbol{q}}_{v_M} = \min\{\tfrac{\boldsymbol{q}_M-\boldsymbol{q}}{\Delta T},\ \boldsymbol{v}_M,\ \sqrt{2\boldsymbol{a}_M(\boldsymbol{q}_M-\boldsymbol{q})},\} \tag{4.33}$$

where ΔT is the time interval of the discrete controller and $\min\{\bullet\}$ and $\max\{\bullet\}$ is the component-wise minimum, respectively maximum of the input vectors. See for example [41] for more details. These new velocity bounds can be used to limit the optimization variable in the QP directly, serving effectively as the highest priority joint-level safety bounds.

The formulation (4.21)–(4.23) is now augmented with the additional inequality constraint on the optimization variable

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}||\boldsymbol{s}||_2^2 + \frac{1}{2}\rho||\dot{\boldsymbol{q}}_v||_2^2 && (4.34)\\ \text{s.t.} \quad & \dot{\boldsymbol{\beta}}_{i_m} \leq \boldsymbol{A}_i\dot{\boldsymbol{q}}_v - \boldsymbol{s} \leq \dot{\boldsymbol{\beta}}_{i_M} && (4.35)\\ & \dot{\bar{\boldsymbol{\beta}}}_{i-1_m} + \bar{\boldsymbol{s}}_{i-1}^* \leq \bar{\boldsymbol{A}}_{i-1}\dot{\boldsymbol{q}}_v \leq \dot{\bar{\boldsymbol{\beta}}}_{i-1_M} + \bar{\boldsymbol{s}}_{i-1}^*, && (4.36)\\ & \dot{\boldsymbol{q}}_{v_m} \leq \dot{\boldsymbol{q}}_v \leq \dot{\boldsymbol{q}}_{v_M}. && (4.37) \end{aligned}$$

One should mention an issue which may arise with such box constraints, which is depicted in Fig. 4.8. A continuously changing task could result in discontinuous joint velocities. However, as acceleration limits are specified as well, these discontinuities are usually tolerable.

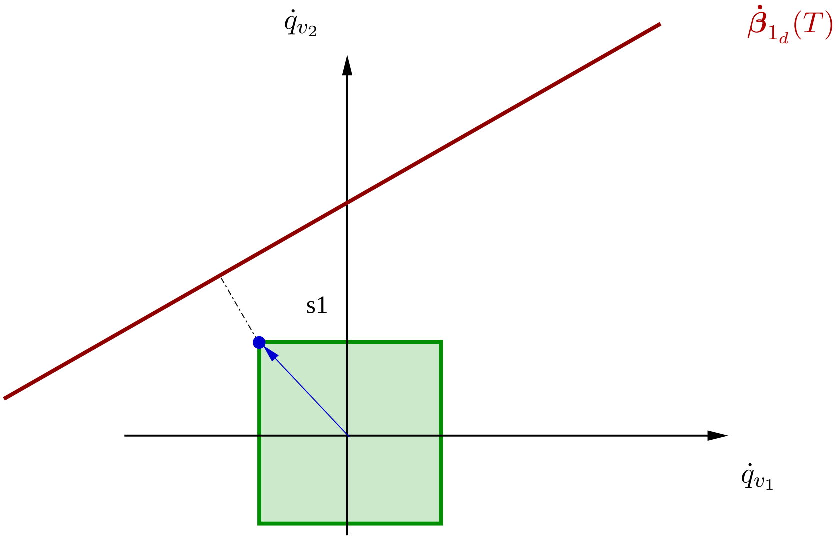


Figure 4.8. Potential issues when implementing joint limits as box constraints. A continuously changing task could result in discontinuous joint velocities.

4.1.4. Quadratic Programming Problem Formulation

The classic QP problem statement is to find a vector $\boldsymbol{w}$, that minimizes a quadratic cost function, subject to linear equality and inequality constraints:

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{H}\boldsymbol{w} + \boldsymbol{a}^T\boldsymbol{w} && (4.38) \\ \text{s.t.} \quad & \boldsymbol{C}\boldsymbol{w} \leq \boldsymbol{b} && (4.39) \\ & \boldsymbol{E}\boldsymbol{w} = \boldsymbol{d}, && (4.40) \end{aligned}$$

where $\boldsymbol{H}, \boldsymbol{C}$ and $\boldsymbol{E}$ are matrices and $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{d}$ are vectors of appropriate size.

In robot control, lower and upper bounds on the task variables are usually imposed. Therefore it is easier to state the inequality constraints as box constraints.

$$\boldsymbol{b}_m \leq \boldsymbol{C}\boldsymbol{w} \leq \boldsymbol{b}_M, \tag{4.41}$$

where $\boldsymbol{b}_m$ is the lower and $\boldsymbol{b}_M$ the upper bound on $\boldsymbol{C}\boldsymbol{w}$. Also, most QP solvers take directly lower and upper bounds ($\boldsymbol{w}_m$ and $\boldsymbol{w}_M$) on $\boldsymbol{w}$ instead of having to formulate them as inequality constraints. With this,the equivalent problem can be stated:

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{H}\boldsymbol{w} + \boldsymbol{a}^T\boldsymbol{w} && (4.42) \\ \text{s.t.} \quad & \boldsymbol{b}_m \leq \boldsymbol{C}\boldsymbol{w} \leq \boldsymbol{b}_M && (4.43) \\ & \boldsymbol{w}_m \leq \boldsymbol{w} \leq \boldsymbol{w}_M. && (4.44) \end{aligned}$$

This corresponds also to the format, which is accepted by the QP solver used in the present implementation [38]. However, the different formulations are completely interchangeable.

To comply with (4.42) – (4.44), $\boldsymbol{w} = [\dot{\boldsymbol{q}}_v \quad \boldsymbol{s}]^T$ is defined and the corresponding QP problem is formulated:

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{H}\boldsymbol{w} && (4.45) \\ \text{s.t.} \quad & \dot{\boldsymbol{\beta}}_m \ \leq \ [\boldsymbol{A} \quad -\boldsymbol{I}_m]\boldsymbol{w} \ \leq \ \dot{\boldsymbol{\beta}}_M && (4.46) \\ & \boldsymbol{w}_m \ \leq \ \boldsymbol{w} \ \leq \ \boldsymbol{w}_M && (4.47) \end{aligned}$$

with

$$\boldsymbol{H} = \begin{bmatrix} \rho \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_m \end{bmatrix} \tag{4.48}$$

$$\boldsymbol{w}_m = \begin{bmatrix} \dot{\boldsymbol{q}}_{v_m} \\ -\infty \end{bmatrix} \tag{4.49}$$

$$\boldsymbol{w}_M = \begin{bmatrix} \dot{\boldsymbol{q}}_{v_M} \\ \infty \end{bmatrix}. \tag{4.50}$$

Here, no bounds on the slack variable are imposed. If a certain task accuracy is required, these bounds can be adjusted accordingly.

This states a reformulation of (4.34)–(4.37) in standard QP form (4.42) – (4.44) resulting in the general recursive formulation with $i = [1 \dots k]$

$$\text{min.} \quad \frac{1}{2}\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} \tag{4.51}$$

$$\text{s.t.} \quad \boldsymbol{b}_{i_m} \;\leq\; \boldsymbol{C}_i \boldsymbol{w} \;\leq\; \boldsymbol{b}_{i_M} \tag{4.52}$$

$$\boldsymbol{w}_m \;\leq\; \boldsymbol{w} \;\leq\; \boldsymbol{w}_M \tag{4.53}$$

with

$$\boldsymbol{b}_{i_m} = \begin{bmatrix} \dot{\boldsymbol{\beta}}_{i_m} \\ \bar{\boldsymbol{b}}_{i-1_m} \end{bmatrix} \tag{4.54}$$

$$\boldsymbol{b}_{i_M} = \begin{bmatrix} \dot{\boldsymbol{\beta}}_{i_M} \\ \bar{\boldsymbol{b}}_{i-1_M} \end{bmatrix} \tag{4.55}$$

$$\boldsymbol{C}_i = \begin{bmatrix} \boldsymbol{A}_i & -\boldsymbol{I}_m \\ \bar{\boldsymbol{A}}_{i-1} & \boldsymbol{0} \end{bmatrix} \tag{4.56}$$

and the initial values

$$\bar{\boldsymbol{A}}_0 = \boldsymbol{0}_{1\times n} \tag{4.57}$$

$$\bar{\boldsymbol{b}}_{0_m} = 0 \tag{4.58}$$

$$\bar{\boldsymbol{b}}_{0_M} = 0. \tag{4.59}$$

For the next iteration

$$\bar{\boldsymbol{b}}_{i_m} = \begin{bmatrix} \dot{\boldsymbol{\beta}}_{i-1_m} + \boldsymbol{s}^*_{i-1} \\ \bar{\boldsymbol{b}}_{i-1_m} \end{bmatrix} \tag{4.60}$$

$$\bar{\boldsymbol{b}}_{i_M} = \begin{bmatrix} \dot{\boldsymbol{\beta}}_{i-1_M} + \boldsymbol{s}^*_{i-1} \\ \bar{\boldsymbol{b}}_{i-1_M} \end{bmatrix} \tag{4.61}$$

$$\bar{\boldsymbol{A}}_i = \begin{bmatrix} \boldsymbol{A}_i \\ \bar{\boldsymbol{A}}_{i-1} \end{bmatrix} \tag{4.62}$$

This approach is summarized in Algorithm C.3 in appendix C.

4.1.5. General Task Specification and Robot Parameterization

The task specification parameters from Sec. 3.2.5 are extended to support the specification of inequality tasks by declaring:

- the task type (or task Jacobian $\boldsymbol{A}$)
- the lower and upper bounds for the task variable $\boldsymbol{\alpha}_m$ and $\boldsymbol{\alpha}_M$ (or $\boldsymbol{\alpha}_d$)
- the feedforward differential task variables $\dot{\boldsymbol{\alpha}}_{m_{\text{ff}}}$ and $\dot{\boldsymbol{\alpha}}_{M_{\text{ff}}}$
- the convergence factor $\boldsymbol{\Lambda}$
- the subspace matrix $\boldsymbol{S}$
- id of the kinematic chain (for kinematic trees)

In general the same $\boldsymbol{\Lambda}$ is chosen for the upper and lower limits.

For the robotic platform the following parameters need to be known:

- the kinematic parameters
- optionally the joint position, velocity, acceleration and torque limits
- the IFC joint stiffness $\boldsymbol{K}$.

This is a very general abstraction, which allows application on a broad range of platforms.

4.2. Passivity Based Stability Proof

While it was shown in multiple works that the interconnection of an IFC block and a robot is stable for external generation of set-points (e.g. [45, 112]), this is not necessarily the case if the SPG block is depending on the IFC or robot state, hence connected via a feedback loop to the system (see Fig. 1.3).

As the solution of the QP problem can not be stated in a closed form, classic stability analysis from control theory can not be applied here. To overcome this, solutions from a domain where similar problems are faced were investigated, namely teleoperation. Here the command input to the robot can not be formally determined as it is received over a usually unreliable communication interface (e.g. Internet). Passivity theory has proved as a useful tool to handle such conditions.

4.2.1. Passivity and Stability

Being a sufficient stability condition, passivity is an intuitive approach to stabilize an nonlinear, partially unknown system. Instead of relying on a model, passivity theory makes assumptions on energetic properties to derive stability conditions. More precisely, if the system dissipates more energy than it generates, it is considered passive, hence stable [98]. Furthermore, a network of passive systems is also passive, what holds for direct as well as for feedback connections. Every subsystem in the network is connected to another by so called power ports, through which energy is exchanged. It is sufficient to observe the energy flow of a system through these ports to make conclusions about its stability.

The energy transmitted to a general M-port system (see Fig. 4.9) is the sum of the individual port energies:

$$E_H = \sum_{j=1}^{M} E_j. \tag{4.63}$$

The overall energy transmitted to a port-system network is the sum of the individual subsystem energies:

$$E = \sum_{j=1}^{M} E_{H_j}. \tag{4.64}$$

In Fig. 4.10 an arbitrary network with one open port is depicted. One can easily recognize that when applying (4.64) and assuming a loss-less connection, the energy transmission

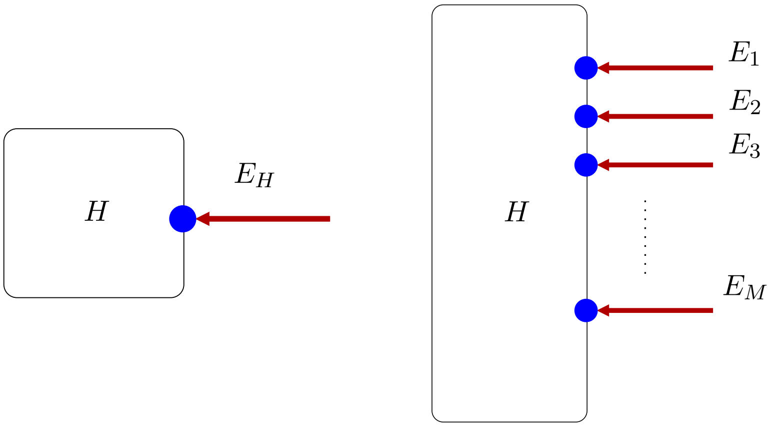


Figure 4.9. One- and M-port system. By convention, an arrow pointing in corresponds to a positive energy transmission E.

is fully defined via the open ports. The system is energetically passive if the energy transmitted to the system is bounded by a constant $c \in \mathbb{R}$, which depends on the initial energy:

$$E \geq -c^2. \tag{4.65}$$

A drawback of the passivity approach is that it could result in an overly conservative controller.

One approach which makes use of the port-model is the passivity observer / passivity controller or time-domain passivity control concept presented in [94, 95, 44], where the commanded torque, respectively velocity, is damped once a violation of the passivity condition is observed. A major drawback of this approach is that it follows a "see-then-action" policy [60], what may be the cause for the noisy behaviour of this strategy. Another approach is the energy bounding algorithm [58], which modulates the control force to comply with the passivity conditions. In [60] a *passive set-position modulation* (PSPM) for a Cartesian impedance controlled manipulator is proposed. As this setup is very similar to the present scenario, the basic concepts are applied to formulate a stability proof. To make the system passive, Lee et.al. propose to implement a virtual energy reservoir, which stores the dissipated energy and use it to execute non-passive actions. This is realized by augmenting the overall system with a virtual energy storage and modulate the desired set-points to the IFC so that the energy transfered to the virtual spring is limited by the amount of energy left in the reservoir. The PSPM is conceptually equal to the usage of the so called energy tanks [16, 37].

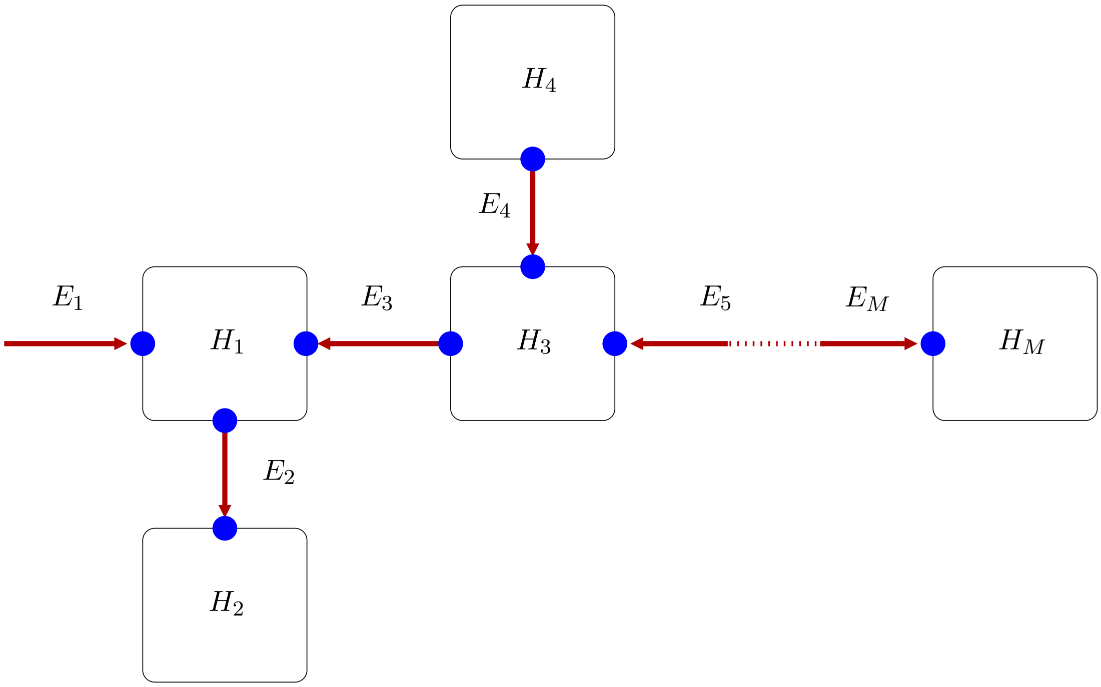


FIGURE 4.10. An arbitrary network port-systems. The overall energy transmission can be observed through the open port only.

4.2.2. Passive Set-Point Modulation

4.2.2.1. Energetic System Analysis

Lets analyze the energy flow for the IFC-SPG system of an individual joint:

$$\tau = K\Delta q - D\dot{q} + \tau_g \tag{4.66}$$

where $\Delta q = (q_v - q)$ and K and D are the corresponding diagonal entries in the stiffness and damping matrices from (2.20). Specific joint indexing is omitted in the following for the sake of readability. The energy flow between the SPG, IFC and the robot is depicted in Fig. 4.11, where the robot is regarded as a system with $n+1$ ports. One for each joint connected to its IFC and one for the connection of the robot to the environment. It can be shown that the robot possesses open loop energetic passivity

$$\int_0^T (\boldsymbol{\tau} - \boldsymbol{\tau}_{ext})^T \dot{\boldsymbol{q}} dt \geq -\frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}\dot{\boldsymbol{q}}. \tag{4.67}$$

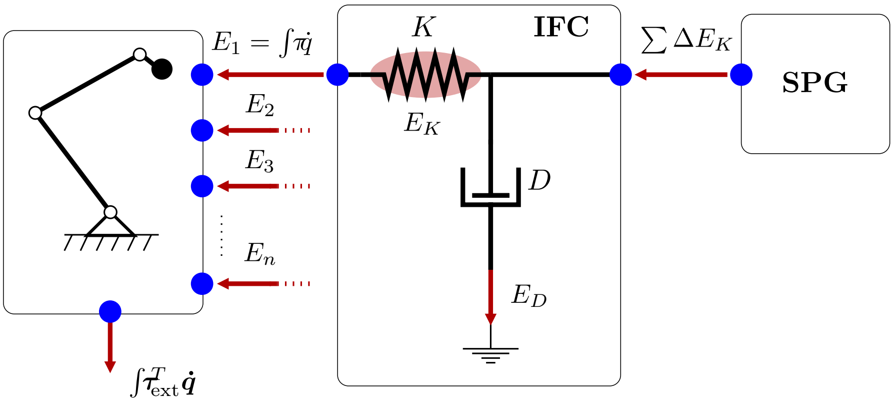

$E_1 = \int \tau \dot{q}$
E_2
E_3
E_n
K
IFC
E_K
D
E_D
$\sum \Delta E_K$
SPG
$\int \boldsymbol{\tau}_{\text{ext}}^T \dot{\boldsymbol{q}}$

Figure 4.11. The energy flow between the robot, IFC and SPG. The energy increment ΔE_K generated by the SPG is a potentially passivity-violating action.

From Fig. 4.11 and (4.65) it can be recognized that

$$-\int_0^T \tau \dot{q} dt \geq -c^2 \tag{4.68}$$

has to hold for every joint to prove passivity of the SPG-IFC block.

The potential energy stored in the virtual spring is

$$E_K = \frac{1}{2} K \Delta q^2 \tag{4.69}$$

and the energy dissipated by the damper after time T is

$$E_D(t) = \int_0^T D \dot{q}^2 dt. \tag{4.70}$$

It is assumed either that the internal IFC controller runs at a high frequency or a physical spring is installed so that (4.66) is continuous. The SPG on the other hand, is regarded as a discrete system, which provides set-points with the frequency $\frac{1}{\Delta T}$. Hence the IFC input $\dot{q}_v$ is equivalent to setting a joint position increment δq_v with the sampling time ΔT:

$$\delta q_v = \dot{q}_v \Delta T \tag{4.71}$$

By taking the difference of the potential energy before and after the position increment, the energy-increase at the discrete sampling point i due to the set-point setting can be computed:

$$\Delta E_K[i] = \frac{1}{2}K((\Delta q[i] + \delta q_v)^2 - \Delta q[i]^2), \tag{4.72}$$

where $\bullet[i]$ denotes the respective quantity at the discrete time-step i. The sign of $\Delta E_K[i]$ is not determined, hence the set-point increment is a potentially passivity-violating action. Besides that, the dissipated energy during the time interval ΔT by the damper can be written as

$$\Delta E_D[i] = \int_{t_i}^{t_i+\Delta T} D\dot{q}^2 dt \tag{4.73}$$

where t_i is the continuous time at the discrete sample point i.

With $E_K(t)$ being the energy stored in the spring at time t, the energy equation can be stated as

$$-\int_0^T \tau\dot{q}dt - E_D(T) + \sum_{j=1}^{i} \Delta E_K[j] = E_K(T) - E_K(0) \tag{4.74}$$

and with (4.68) the IFC-SPG block for one joint is passive with respect to the input-output-pair $\{\tau, -\dot{q}\}$ if

$$-\int_0^T \tau\dot{q}dt = E_K(T) - E_K(0) + E_D(T) - \sum_{j=1}^{i} \Delta E_K[j] \geq -c^2 \tag{4.75}$$

Starting with the energy storage function

$$E_r[i] = E_r[i-1] + E_D(t_i) - \Delta E_K[i], \tag{4.76}$$

for practical reasons the following adjustments have to be done according to [60]:

1. E_D can not be computed due to missing information on the joint velocity between the sampling points, therefor the minimum damped energy $E_{D_{\min}}[i]$ has to be used instead. By applying the Caushy-Schwartz inequality [1]

$$\int_t^{t+\Delta T} |\dot{q}|dt \leq \sqrt{\int_t^{t+\Delta T} |\dot{q}|^2 dt \times \int_t^{t+\Delta T} 1dt} \tag{4.77}$$

it can be shown that

$$\Delta E_D[i] = \int_{t_i}^{t_{i+1}} D\dot{q}^2 dt \geq \frac{D}{\Delta T}(q_{\max}[i] - q_{\min}[i])^2 := \Delta E_{D_{\min}}[i] \tag{4.78}$$

where $q_{\max}[i] = \max(q(t_i), q(t_{i+1}))$ and $q_{\min}[i] = \min(q(t_i), q(t_{i+1}))$, assuming a monotonic $\dot{q}$ during ΔT. Since computing $\Delta E_{D_{\min}}[i]$ requires future information, i.e. evolution of $\dot{q}$ during $[t_i, t_{i+1}]$ the dissipated energy during the previous time step is used instead. This is still valid, since as $E_D(t_i) \geq E_D(t_{i-1})$. Therefore $E_{D_{\min}}[i-1]$ replaces $E_D(t_i)$ in (4.76).

2. Too extensive energy storage might lead to aggressive behavior while theoretically still being passive. Therefore a maximum capacity $E_{r_{\max}}$ for E_r is defined, what is called energy ceiling in [60].

3. For highly dissipative environments it is useful to transfer some energy to the system. This action also makes sure that the robot does not get stuck when all the energy in the storage is depleted. This is realized by adding the shuffling term $\Delta E_{\text{shuffle}}$ to the energy storage function (4.76).

4. With a depleted energy reservoir at $t = 0$ no motion would be possible in the beginning of the task, since any extension of the virtual spring would result in a violation of the passivity condition. To overcome this "take-off" problem, the storage is initialized with the energy $E_r[0] = E_{r_{\text{init}}}$.

The new energy storage function with minimum dissipated energy and shuffling term is

$$E_r[i] = E_r[i-1] + \Delta E_{D_{\min}}[i-1] + \Delta E_{\text{shuffle}} - \Delta E_K(t_i) \tag{4.79}$$

By requesting $E_r[i] \geq 0$, it is assured that the energy generated by the SPG is not larger than the energy dissipated in the IFC. This limitation can be directly incorporated into the existing SPG framework by adjusting the velocity limits on $\dot{q}_v$. From (4.71) and (4.72), velocity limits can be derived, which have to be obeyed in addition to (4.32) and (4.33):

$$\dot{q}_{p_m} = \frac{1}{\Delta T}\left(-\sqrt{\frac{2}{K}(E_r[i-1] + \Delta E_{D_{\min}}[i-1] + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q\right) \quad f. \quad \Delta q \leq 0 \tag{4.80}$$

$$\dot{q}_{p_M} = \frac{1}{\Delta T}\left(\sqrt{\frac{2}{K}(E_r[i-1] + \Delta E_{D_{\min}}[i-1] + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q\right) \quad f. \quad \Delta q \geq 0 \tag{4.81}$$

The new energy flow diagram including the energy tank is depicted in Fig. 4.12

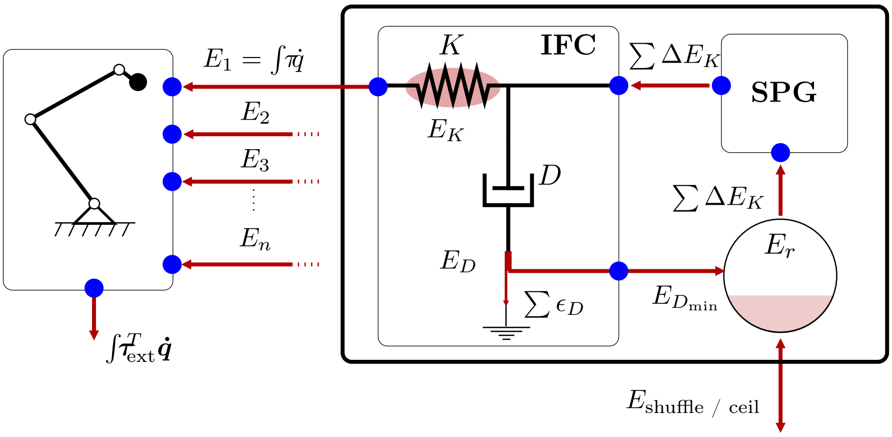


FIGURE 4.12. The energy flow with the augmented energy reservoir. By coupling the energy generation of the SPG to the virtual energy reservoir, the overall system is guaranteed to be passive, hence stable.

4.2.2.2. Stability Analysis

By assuming that the energy transfered to the system via $\Delta E_{\text{shuffle}}$ is bounded by a positive constant

$$b^2 \geq \sum_{j=1}^{N} \Delta E_{\text{shuffle}}, \tag{4.82}$$

the bounding constant is defined as

$$c^2 = E_K(0) + E_{r_{\text{init}}} + b^2. \tag{4.83}$$

Since $\Delta E_{\text{shuffle}}$ is user defined 4.82 is always true. Now using (4.79) and (4.74) it follows that

$$\begin{aligned} -\int_0^T \tau \dot{q} dt = & E_K(T) - E_K(0) + \sum_{j=1}^{i} [\Delta E_{D_{\min}}[j-1] - \Delta E_K[j] + \epsilon_D[j-1]] + \int_{t_i}^{T} D\dot{q}^2 dt \\ \geq & E_K(T) - E_K(0) + \sum_{j=1}^{i} [E_r[j] - E_r[j-1] - \Delta E_{\text{shuffle}}] \\ \geq & E_K(T) - E_K(0) - E_{r_{\text{init}}} + E_r[i] - \sum_{j=1}^{i} \Delta E_{\text{shuffle}} \geq -c^2, \end{aligned} \tag{4.84}$$

where $\epsilon_D[i] = \Delta E_D[i] - \Delta E_{D_{\min}}[i-1] \geq 0$ is the energy leak due to the conservative estimation of the dissipated energy.

4.3. Experimental Verification

The frequency of the discrete controller is $500Hz$ and the stiffness is set to $\boldsymbol{K} = 200\frac{Nm}{\text{rad}}\boldsymbol{I}_7$ for all experiments if not stated otherwise. The task convergence factors $\boldsymbol{\Lambda}$ and the regularization factor $\rho = 0.01$ where chosen heuristically. The C++ QP library *qpOASES* [38] was used to carry out the optimization.

The considered joint-level limits are summarized in table B.1. The limits on $\boldsymbol{q}_v$, $\dot{\boldsymbol{q}}_{\boldsymbol{v}}$ and $\ddot{\boldsymbol{q}}_{\boldsymbol{v}}$ are implemented as explicit constraints on the optimization variable $\dot{\boldsymbol{q}}_v$ as shown in Sec. 4.1.3.2. The limits on the joint torques $-\boldsymbol{\tau}_{s_M} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{s_M}$ are implemented as an inequality task at the highest priority-level and is not explicitly stated in the task descriptions of the experiments.

Constrained Trajectory Tracking First, a generic mixed components task is executed, which is following a Cartesian trajectory on a table while applying a constant force on it. In addition, there is an obstacle blocking the way. This task is a placeholder for different mixed component tasks with uncertain task geometry and the possibility for uncertain collisions, which are common in service robotics tasks. The setup is depicted in Fig. 4.13 and the main subtasks are summarized in table 4.1. Three different task descriptions were applied to demonstrate some basic properties of the approach and highlight the simplicity of the task programming interface. According to table 4.1 tasks 2,4 and 5 are the same for all setups, which are:

- maintaining the initial orientation of the end-effector $\boldsymbol{o}_{\text{init}}$ (task 2)
- following a time dependent translational trajectory with the end-effector (task 4)
- maximizing the distance of the joints to their limits (task 5)

First, the force task is implemented as an equality constraint along the x-axis only. Then the torque limits from table B.1 are lowered to avoid large interaction torques (task 1b). Finally, the force task is defined as an equality in its x-component and as an inequality in the y and z components to limit the interaction forces explicitly (task 3a). The results are depicted in Fig. 4.14.

Table 4.1. Set of subtasks for constrained trajectory tracking

task	type	$\boldsymbol{\alpha}_m$	$\boldsymbol{\alpha}_M$	$\boldsymbol{S}$
1a	joint torque	$-50Nm\mathbf{1}_{7\times1}$	$50Nm\mathbf{1}_{7\times1}$	$\boldsymbol{I}_7$
1b	joint torque	$-5\mathbf{1}_{7\times1}$	$5Nm\mathbf{1}_{7\times1}$	$\boldsymbol{I}_7$
2	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix}\mathbf{0}_3\\ \boldsymbol{I}_3\end{bmatrix}$
3a	wrench	$10N$	$10N$	$[1\ 0\ 0\ 0\ 0\ 0]^T$
3b	wrench	$\begin{bmatrix}10\\ -10\\ -10\end{bmatrix}N$	$\begin{bmatrix}10\\ 10\\ 10\end{bmatrix}N$	$\begin{bmatrix}\boldsymbol{I}_3\\ \mathbf{0}_3\end{bmatrix}$
4	cart. pose	$\boldsymbol{p}_d(t)$	$\boldsymbol{p}_d(t)$	$\begin{bmatrix}\boldsymbol{I}_3\\ \mathbf{0}_3\end{bmatrix}$
5	joint position	$\mathbf{0}$	$\mathbf{0}$	$\boldsymbol{I}_7$

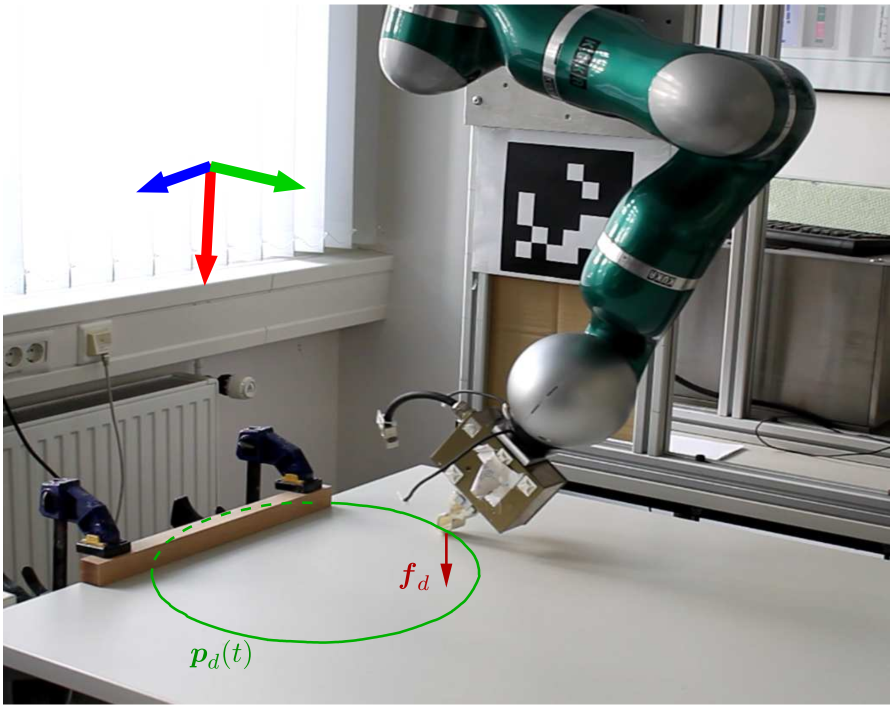


Figure 4.13. Experimental setup for constrained trajectory trajectory. A sample mixed components task with an unexpected obstacle. The main subtasks are to apply a constant interaction force and follow a circular trajectory. Different strategies to deal with unexpected obstacles are compared: relying on the IFC compliance, limiting the joint torques and limiting the end-effector forces.

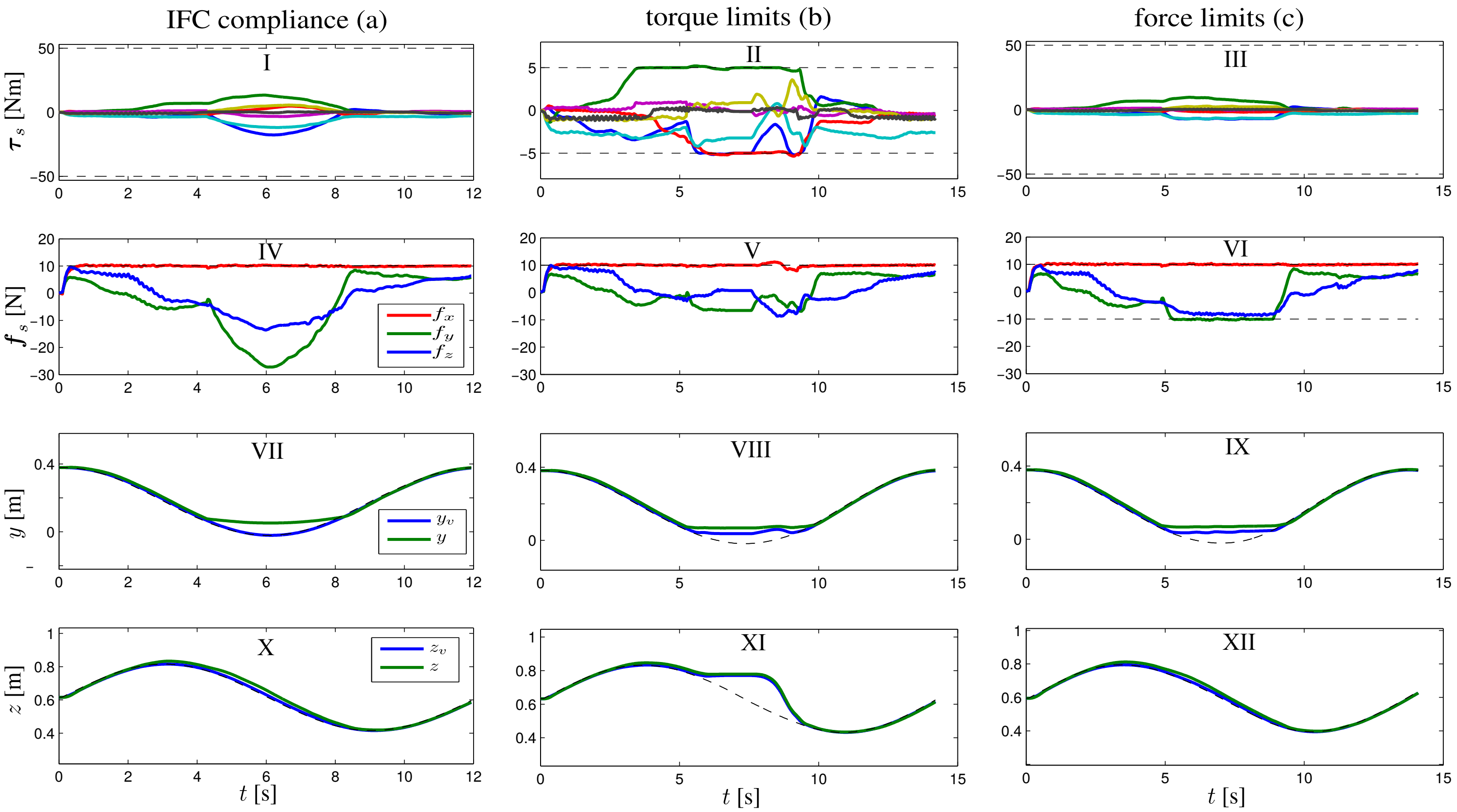


Figure 4.14. Each column shows the propagation of the relevant task variables for three different task specifications from Table 4.1. For the trajectory following task, both the virtual and the physical position of the end effector are plotted. In each graph the dashed line represents the task constraints (desired value or limits).
Left: the obstacle impedes the motion of the physical end effector along the y-axis (VII). This is compensated by the inherent compliance of the IFC, leading to additional interaction forces in the y- and z-directions (IV). Forces along y and z are not constrained by the task specification. This approach corresponds basically to a classical IFC application.
Middle: the torque limits are lowered to ±5Nm in order to avoid large interaction torques (II). This prevents the virtual end effector from penetrating further into the obstacle (VIII). However, due to friction, in this case the new torque limits are so conservative that at some point no motion is possible at all while the virtual end-effector penetrates the obstacle (XI).
Right: another possibility is to explicitly limit the interaction forces instead of the torques, making it possible to tune the allowed forces in Cartesian space, to account for expected resistance, e.g. friction (VI).

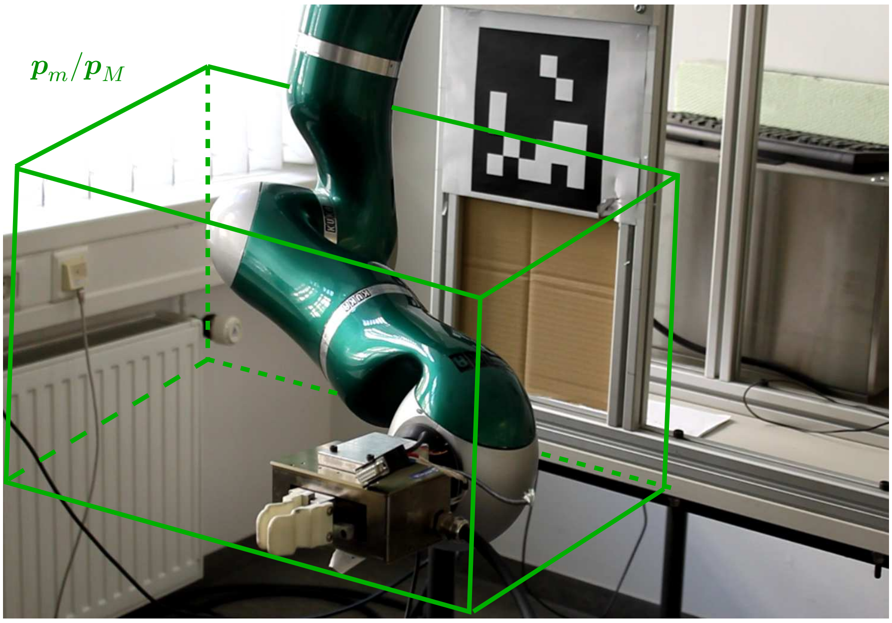


Figure 4.15. Keeping the orientation constant while constraining the end-effector position to a bounding box while minimizing the joint torques.

Orientation Control with Position Boundaries The next set of tasks demonstrates different behaviors, which could be achieved in a simple way. The main task is to maintain the initial orientation of the end-effector while minimizing the torques (see table 4.2, tasks 1 and 3). With these two tasks active, the robot can be pushed around manually, while keeping the initial orientation and obeying the joint limits from table B.1. This task corresponds to the cup-holding task from Chapter 3. By inserting an inequality task for the position (task 2 in table 4.2), the virtual end-effector can be restricted to stay within a certain bounding box (see Fig. 4.15). The experiment was carried out with low and high stiffness parameters and the results are plotted in Fig. 4.16.

By making the positioning task an equality constraint and adding an inequality force task, the end-effector of the robot can be programmed to stay in a certain pose as long as the force limits are not violated. When the external forces vanish, the end-effector returns to its initial pose. The associated task description and plots can be found in table 4.3 and Fig. 4.17.

The next task (table 4.4) is to hold a constant position in all directions besides the end-effector z-axis. The $\boldsymbol{S}$ matrix is first defined as a selection matrix in end-effector coordinates and then transformed to the root frame by premultiplication with the block diagonal matrix $\operatorname{diag}(\boldsymbol{R}_{\text{ee}}, \boldsymbol{R}_{\text{ee}})$, where $\boldsymbol{R}_{\text{ee}}$ is the rotation matrix of the end-effector. This $\boldsymbol{S}$ is then used to transform the desired end-effector velocity $\dot{\boldsymbol{x}}_{\boldsymbol{d}} = \boldsymbol{\Lambda}_x(\boldsymbol{x}_{\text{init}} - \boldsymbol{x}_v)$ in $\boldsymbol{S}$-coordinates.

Table 4.2. Set of subtasks for maintaining a constant orientation with optional Cartesian inequality constraint

task	type	$\boldsymbol{\alpha}_m$	$\boldsymbol{\alpha}_M$	$\boldsymbol{S}$
1	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix}\mathbf{0}_3\\ \boldsymbol{I}_3\end{bmatrix}$
2	cart. pose	$\begin{bmatrix}0.2\\0.2\\0.4\end{bmatrix} m$	$\begin{bmatrix}0.5\\0.5\\0.7\end{bmatrix} m$	$\begin{bmatrix}\boldsymbol{I}_3\\ \mathbf{0}_3\end{bmatrix}$
3	joint torque	$\mathbf{0}$	$\mathbf{0}$	$\boldsymbol{I}_7$

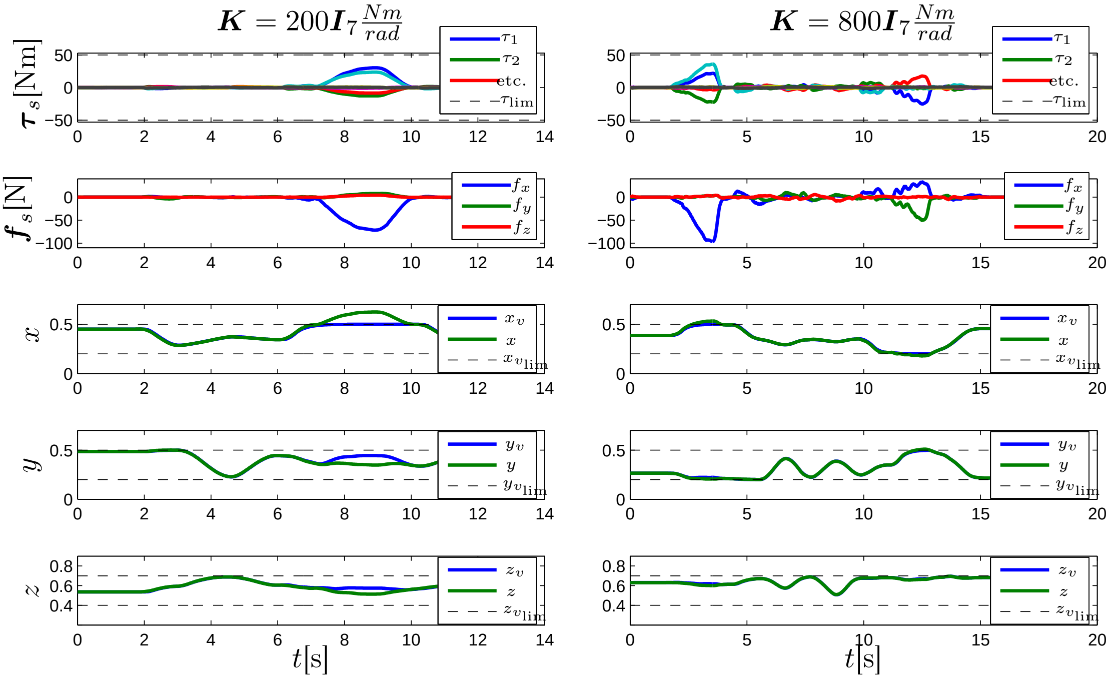


Figure 4.16. Propagation of the main subtasks for maintaining a constant end-effector orientation while minimizing the external torques and Cartesian inequality constraint. While inside the bounding box, the robot can be pushed around manually. Due to the inherent compliance of the IFC it is also possible to push the physical end-effector out of the box while increasing the interaction torques. The stiffness parameter of the IFC determines the amount of force required to push the end-effector out of the box.

Table 4.3. Keeping the end-effector pose with static wrench constraints

task	type	$\boldsymbol{\alpha}_m$	$\boldsymbol{\alpha}_M$	$\boldsymbol{S}$
1	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\boldsymbol{o}_{\text{init}}$	$\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$
2	wrench	$\begin{bmatrix} -10 \\ -10 \\ -10 \end{bmatrix} N$	$\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} N$	$\begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0}_3 \end{bmatrix}$
3	cart. pose	$\boldsymbol{p}_{\text{init}}$	$\boldsymbol{p}_{\text{init}}$	$\begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0}_3 \end{bmatrix}$
4	joint torque	$\boldsymbol{0}$	$\boldsymbol{0}$	$\boldsymbol{I}_7$

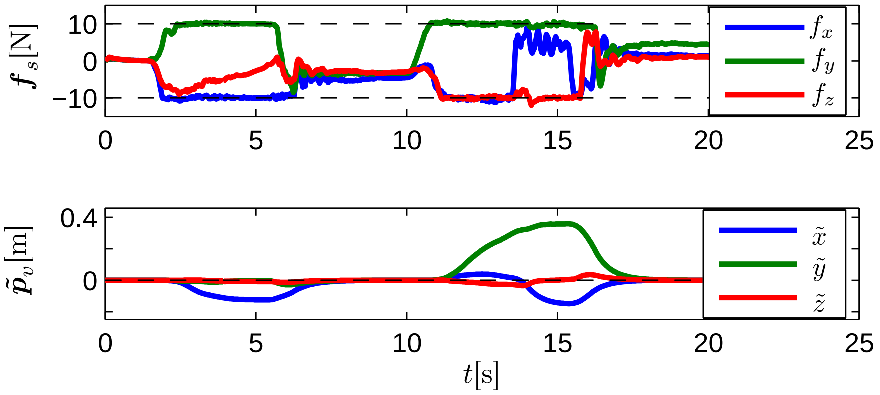


Figure 4.17. The interaction forces $\boldsymbol{f}_s$ and position error $\tilde{\boldsymbol{p}}_v$ for the task specification from table 4.3 while applying external forces on the end-effector manually. When the force limits are reached, the end-effector deviates from its desired position and returns to it when the external forces vanish.

Table 4.4. Set of subtasks for holding a constant pose while allowing motion along end-effector z-axis

task	type	$\boldsymbol{\alpha}_d$	$\dot{\boldsymbol{\alpha}}_{\text{ff}}$	$\boldsymbol{S}$
1	cart. pose	NA	$\boldsymbol{S}_1^T \dot{\boldsymbol{x}}_d$	$\boldsymbol{S}_1 = \text{diag}(\boldsymbol{R}_{\text{ee}}, \boldsymbol{R}_{\text{ee}}) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
2	joint torque	$\boldsymbol{0}$	$\boldsymbol{0}$	$\boldsymbol{I}_7$

Stability To demonstrate the effect of the stabilizer, the following position regulator for joint 5 is considered:

$$\dot{q}_{5_d} = k_p(q_{5_d} - q_{5_v}) + k_d \dot{q}_5, \tag{4.85}$$

with $k_d > 0$. For sufficiently large k_d, the intrinsic damping of the IFC and the mechanism are not enough to keep the motion exponentially stable. The velocity limits (4.80) and (4.81) are used to stabilize the resulting motion. The task specification can be looked up in table 4.5.

The following numerical parameters were used for the IFC and (4.85):

- $\boldsymbol{K} = 200\boldsymbol{I}_7$
- $k_p = 100$
- $k_d = 0.8$

The resulting plots can be looked up in Fig. 4.18, where the stabilized and the non-stabilized case are compared.

TABLE 4.5. Set of subtasks for stability test

task	type	$\boldsymbol{\alpha}_d$	$\dot{\boldsymbol{\alpha}}_{\text{ff}}$	$\boldsymbol{S}$
1	joint position	0	$\dot{q}_{5_d}$	$[0\ 0\ 0\ 0\ 1\ 0\ 0]^T$

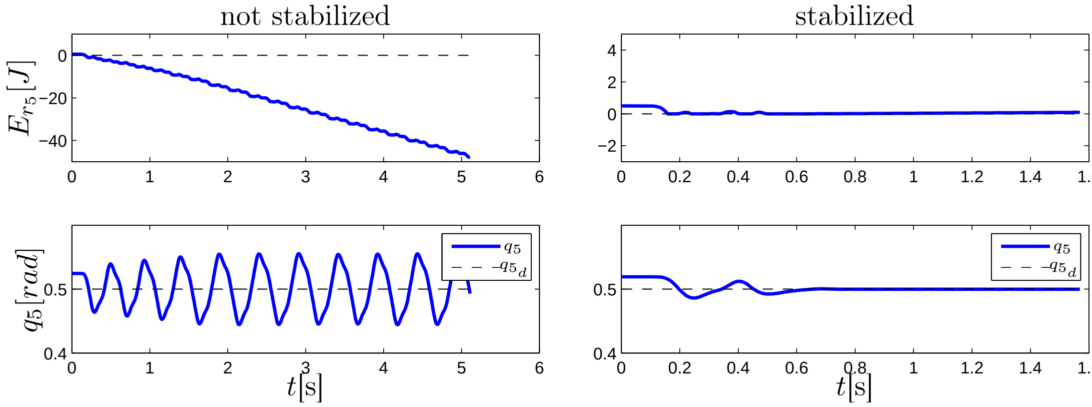


FIGURE 4.18. Energy stored in the reservoir and position error for an unstable joint position task specification. Left: without the additional limits (4.80), respectively (4.81) the energy generation of the SPG is independent from the energy stored in the reservoir. Right: coupling the energy generation to the energy in the reservoir stabilizes the task regulation.

4.4. Concluding Remarks

The main goal of this chapter was to extend the scheme presented so far to support inequality tasks. Finding the virtual joint velocity, according to a set of hierarchical inequality task is formulated as a set of QP optimization problems. For every task in the hierarchy a QP is solved, meeting the specified inequality task as good as possible while obeying the higher priority tasks. This approach was previously proposed on kinematic-level by [51] and applied for the IFC case. The functionality was experimentally verified on a joint impedance controlled robot.

The regulated system was identified as not necessarily stable and a way of enforcing stability was presented by applying the concept of passive set-point modulation. The energy transmitted by the SPG of each joint to the IFC and hence to the robot is coupled to a virtual energy reservoir, which is filled by reharvesting a part of the energy dissipated in the IFC, enforcing passivity of the system. These stability conditions are directly incorporated as additional inequality constraints on the virtual joint velocities without changing the previously derived formalism.

Chapter 5

Constrained Manipulation Using Indirect Force Control

In this chapter, the task of manipulating unknown constrained mechanisms is discussed and a strategy is presented, which accounts for the special nature of the problem is proposed. Three concrete approaches are presented, which have been developed throughout the work for this thesis and which mark important development stages for the final strategy, which takes advantage of the task specification scheme derived in Chapters 3 and 4.

The motivation behind the presented approach and the related work can be found in Sec. 5.1. The general assumptions underlying the proposed strategy and the basic idea is covered in Sec. 5.2. Sec. 5.3 presents the original approach, which was originally developed for an admittance controlled mobile manipulator operating unknown mechanisms. In Sec. 5.4 this approach is modified for application on a robot with joint space compliance and the issue of overlapping subtasks, i.e. force and positioning, is discussed. In Sec. 5.5 the task specification scheme proposed in this thesis is used to separately regulate the force and positioning subtasks involved in the manipulation. Sec. 5.6 concludes the chapter.

5.1. Motivation and Related Work

In recent years, robotics research spread out from the specialized area of industrial plants into everyday human environments, which are designed for human abilities. The integration of robots into human environments is desirably arranged seamlessly without major modifications to the environment itself. Many tasks require the manipulation of simple mechanisms such as doors, drawers or lockers. Robots must be able to operate them in a general manner like any human can do. Actuating such mechanisms is not always affordable, and in fact, human-like abilities is one of the key aspects of humanoid robotics.

A major problem arises from the wide range of the devices described above, which make model-based approaches difficult to apply. There exists a large variety of opening arcs, handles, and general techniques to operate such mechanisms. Besides, the device to operate is not always unresisting. Magnets, springs or tight gaskets keep drawers, respectively doors, closed and require a certain amount of force for operation. The only property all those devices have in common is their restriction of movement to one–DoF trajectories. A drawer can only be pulled or pushed along one direction just as a door swings along a certain arc. A human does not need to know the position of the hinge or the pushing/pulling direction. Instead he just tries to push or pull and lets the structure guide the rest of the motion. Unlike human, very few robotic systems have taken advantage of this fact.

Many approaches for constrained manipulation in unstructured environments focus on modeling the constraint and estimate the model parameters as good as possible, e.g. [86, 92, 87, 25, 75, 93]. Active interaction forces, which are required to move along the constrained trajectory are often not considered and these approaches are limited by the underlying model of the constraint. Also, external sensors like vision or laser scanners are often used, which introduce an additional source of uncertainty and make the complete system often quite complicated.

Approaches which use simple models and rely only on internal sensing (position- and force sensors) often assume orthogonal velocity and force subspaces, neglecting potentially required interaction forces (active forces). Examples of such approaches are [100, 21, 69], where only little active forces are postulated.

In [83] and [88] a desired force has to be predefined, what can be problematic for changing required interaction forces, e.g. for latch mechanisms like a microwave door.

In [48] a compliant manipulator is used to pull open doors and drawers. The control goal is to follow the constraint trajectory with the manipulators end-effector without considering the actually applied interaction forces. The compliance is used to compensate for uncertainties. This method is improved by learning the kinematics of the operated mechanism in [110].

Karayiannidis et al. proposed a velocity controller with force feedback to open revolute and prismatic doors and drawers [52, 53]. While stable execution of the task is proved, the well-known potential risk of contact instability when applying force feedback is not discussed.

5.2. Underlying Assumptions and Basic Idea

The aim of the present work is to provide a strategy for constrained manipulation, which should be as general as possible regarding the applied robotic hardware as well as the operated mechanism. To achieve this, an IFC structure serves as general abstraction layer for the robot. Besides that, it is avoided to use external sensing or force data and rely on a minimal model for the constrained mechanism.

The presented approach is inspired by [83], where the concept of following the path of least resistance is introduced in the context of constraint manipulation to generate direct force commands for a robotic manipulator.

The underlying idea for indirect force controlled robots follows this basic scheme and can be divided into three steps:

1. **Exploration** – Probe the unknown constraint with the virtual manipulator according to the current estimation of the direction of possible motion.
2. **Estimation** – Observe the actual motion of the robot to update the estimation.
3. **Canceling Erroneous Forces** – Projection of the currently applied forces on the estimated direction of motion to nullify drift errors, accumulated during the exploration phase.

In the following the underlying constraint model is presented and the details of step 2 is covered, since it stays the same for the different versions of the approach.

5.2.1. Constraint Modeling

As the present work has a strong practical motivation, certain properties of every day mechanisms and ways of interacting with them are assumed.

- The mechanical system which is operated has only one translational DoF, which applies for many real world mechanisms like doors, drawers or cranks.
- The contact between the end-effector and the mechanism is taking place at some known interaction point, which imposes a bilateral constraint on the translational DoF's of the end-effector.
- The end-effector orientation is, at least locally, unconstrained.

The last assumption is practically motivated, as with most types of general grippers, it is difficult to achieve a full mechanical coupling at the interaction point. Some grasping strategies (e.g. caging grasps [30]) even aim explicitly for relaxing these task constraints. However, the case of a fixed grasp will also be briefly discussed. Hence, the end-effector translational motion is restricted to a one-dimensional trajectory Φ_t in space, which is determined by the mechanism it is interacting with. The type of the predefined trajectory (linear, circular) and its characteristics (arc, direction) are unknown. Thus, for stiff environments, the translational velocity of the end-effector $\dot{\boldsymbol{p}}$ is always aligned with the three-dimensional unit vector $\boldsymbol{d}_p$, denoting the single possible direction of motion.

Orthogonal Decomposition of Interaction Forces An important aspect of the presented approach is, that constraints with dynamic properties are considered, which practically means, that forces need to be applied along $\boldsymbol{d}_p$ to operate the mechanism. Still, it is desired to avoid explicit modeling of the environmental dynamics, since it is very difficult to model every potential case. Instead, interaction forces are decomposed into parallel and perpendicular components with respect to $\boldsymbol{d}_p$. Therefore, the commanded interaction force $\boldsymbol{f}$, generated by the IFC, can be orthogonally decomposed according to

$$\boldsymbol{d}_p \parallel \boldsymbol{f}_{\parallel} = \boldsymbol{P}_p \boldsymbol{f} \tag{5.1}$$

$$\boldsymbol{d}_p \perp \boldsymbol{f}_{\perp} = (\boldsymbol{I}_3 - \boldsymbol{P}_p)\boldsymbol{f} \tag{5.2}$$

$$\boldsymbol{f} = \boldsymbol{f}_{\perp} + \boldsymbol{f}_{\parallel}, \tag{5.3}$$

where the 3×3 projection matrix $\boldsymbol{P}_p = \boldsymbol{d}_p \boldsymbol{d}_p^T$ maps any three-dimensional column vector on $\boldsymbol{d}_p$. Fig. 5.1 depicts this.

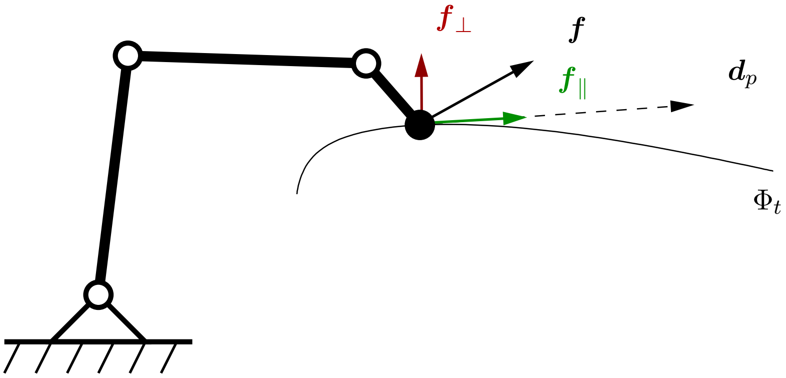


Figure 5.1. Orthogonal decomposition of interaction forces.

5.2.2. First-Order Constraint Estimation

$\boldsymbol{d}_p$ corresponds to the first-order constraint parameter, which is the tangent of Φ_t at the current interaction point. This local model of the constraint covers a broad range of human operated mechanisms. Estimating the geometric parameters of a constrained mechanism is a wide topic of its own and would go beyond the scope of this thesis. An extensive survey can be found in [63]. Since the focus of this thesis does not lie on constraint estimation, only one simple approach, which is used throughout the experiments is described here for completeness. However, any other first-order constraint estimator could be used in the presented manipulation strategy. External sensing as well as force measurements should be avoided, hence only position data is used to obtain the estimation $\hat{\boldsymbol{d}}_p$ of $\boldsymbol{d}_p$.

As proposed by [83], filtering is applied to obtain $\hat{\boldsymbol{d}}_p$ but unlike [83], the translational Cartesian velocity $\dot{\boldsymbol{p}}$ is filtered. For this, a *simple moving average* (SMA) filter is used, which is the unweighted mean of the previous N measurements, where N denotes the order of the filter. The considered time window is then determined via $N\Delta T$. For a discrete filter the SMA is

$$\hat{\boldsymbol{d}}_p = \mathrm{norm}(\frac{1}{N}\sum_{j=0}^{N}\dot{\boldsymbol{p}}[i-j]) \tag{5.4}$$

where $\dot{\boldsymbol{p}}[i]$ is the end-effector translational velocity measurement at time step i. As the normalization is applied after filtering, faster velocities have a bigger impact on the outcome than slower ones, which is a useful property since slower velocities are usually more afflicted by noise.

The smaller N is chosen, the faster the estimation will react to changes of the actual direction. However, the consequential negative effects on stability and increased volatility of the estimate require a conservative choice for N.

Fig. 5.2 shows how a noisy velocity signal is smoothened by the filter, but also the lag, which is introduced in addition. This is a common property of FIR filters as the SMA.

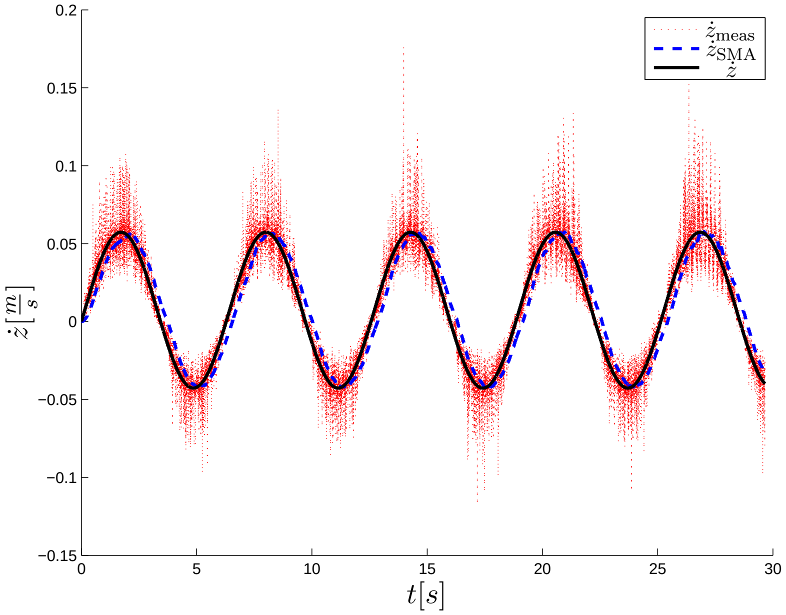


FIGURE 5.2. Effect of moving average filtering with a time window of $0.5s$. The noise in the measured signal $\dot{z}_{\text{meas}}$ is effectively eliminated while a lag in the filtered signal $\dot{z}_{\text{SMA}}$ is introduced.

5.3. Original Approach on an Admittance Controlled Robot

The original approach [68] was developed for an admittance controlled mobile manipulator (see appendix B for details of the used hardware). The Cartesian compliance of this control scheme, results in the nice property, that the pose deviation $\boldsymbol{x}_v - \boldsymbol{x}$ is aligned with the applied interaction wrench $\boldsymbol{h}$[1] (see also Fig. 2.8 in Chapter 2). This holds two major advantages. First, assigning the virtual end-effector position $\boldsymbol{p}_v$ results directly in an interaction force towards $\boldsymbol{p}_v$ and second, the orientation can be regulated independently without affecting the interaction forces.

[1] static and dynamic components

5.3.1. Exploration

For Cartesian compliance, the exploration can be realized by

1. inducing an initial translational velocity of the end-effector $\dot{\boldsymbol{p}}_{\boldsymbol{v}} = \nu \boldsymbol{d}_{p_0}$,
2. applying the first-order estimation to obtain a normalized $\hat{\boldsymbol{d}}_p$,
3. continue the exploration along $\hat{\boldsymbol{d}}_p$ by setting $\dot{\boldsymbol{p}}_{\boldsymbol{v}} = \nu \hat{\boldsymbol{d}}_p$,

with $\boldsymbol{d}_{p_0}$ as the initial estimation and ν as desired exploration velocity. This leads to a build up of interaction forces along $\hat{\boldsymbol{d}}_p$ with a slope of $\boldsymbol{K}_p\nu$ in $\frac{N}{s}$. The physical constraint immediately forces the end-effector on the path of least resistance and the admittance controller compensates for uncertainties.

For practical purposes, the first order constraint estimation (5.4) is extended with a deadband, having the threshold v_{th}. If $||\dot{\boldsymbol{p}}|| < v_{\text{th}}$, the measurement is dropped and some constant vector is assigned to $\dot{\boldsymbol{p}}$, e.g. along some end-effector axis or the last valid estimate:

$$\dot{\boldsymbol{p}} := \begin{cases} \dot{\boldsymbol{p}}, & \text{for} \quad ||\dot{\boldsymbol{p}}|| \geq v_{\text{th}} \\ w_0 \hat{\boldsymbol{d}}_p, & \text{for} \quad ||\dot{\boldsymbol{p}}|| < v_{\text{th}} \end{cases}, \tag{5.5}$$

with $w_0 > 0$ as some constant weighting factor. The main purpose of this action is to reject small, hence noisy velocity measurements from the estimate, but also to initialize the movement along some known direction when no motion is present. In addition, without the deadband, the normalization in (5.4) would become ill-conditioned for small velocities, leading to oscillations and potential failure of the task.

5.3.2. Canceling Erroneous Forces

Many devices in human environments need a certain amount of force to move them or initialize the movement. As a result, $\boldsymbol{p}_v$ may depart from the constraint trajectory in order to apply the required forces or due to an erroneous $\boldsymbol{d}_{p_0}$, before the motion direction can be reliably obtained. Accumulated erroneous forces $\boldsymbol{f}_\perp$ need to be continuously canceled by projecting $\boldsymbol{p}_v$ on the line determined by the direction $\hat{\boldsymbol{d}}_{\boldsymbol{p}}$ and the current interaction point $\boldsymbol{p}$ by

$$\boldsymbol{p}_v := \boldsymbol{p} + \hat{\boldsymbol{P}}_p(\boldsymbol{p}_v - \boldsymbol{p}), \tag{5.6}$$

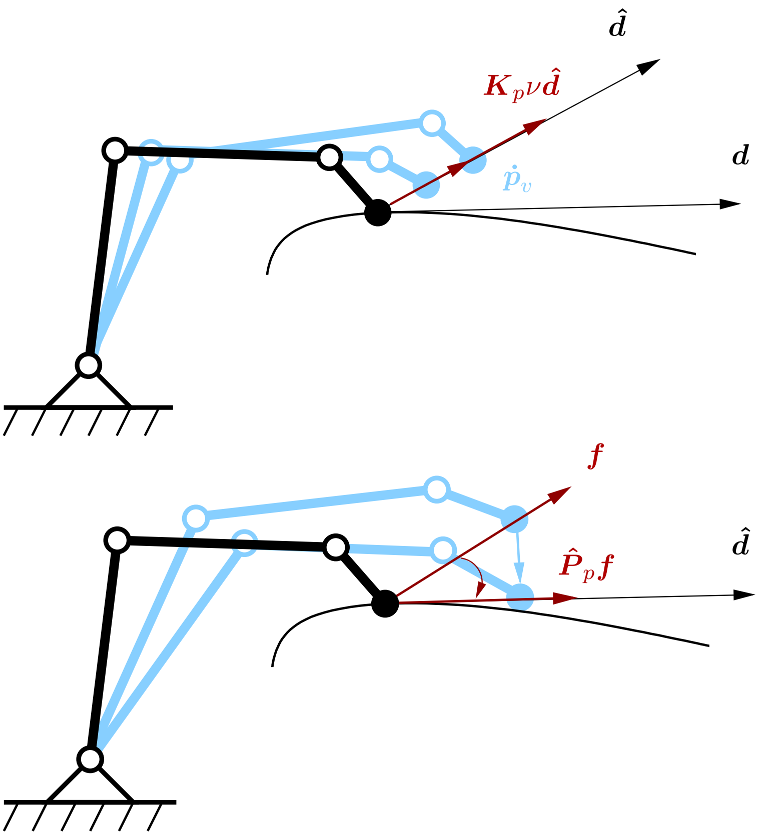


FIGURE 5.3. Constrained manipulation strategy for Cartesian compliance. The end-effector explores the constraint along the estimated direction of possible motion $\hat{\boldsymbol{d}}$ instantaneously. The projection on $\hat{\boldsymbol{d}}$ cancels accumulated erroneous forces.

where $\hat{\boldsymbol{P}}_p = \hat{\boldsymbol{d}}_p\hat{\boldsymbol{d}}_p^T$. A visualization of the exploration and projection stages is given in Fig. 5.3.

5.3.3. Orientation Regulation

The applied interaction forces $\boldsymbol{f}$ and rotational motion are decoupled for a Cartesian admittance controller. Hence orientation can be controlled in any way, e.g. remaining constant or adjusting the end-effector orientation to $\hat{\boldsymbol{d}}_p$, for the assumed loose grasp at the interaction point.

Assuming a Fixed Grasp Given the case that a fixed grasp is realized, the required orientation of the end-effector depends on the evolution of $\boldsymbol{d}$. A simple strategy within the admittance control framework with tunable admittance parameters is to set the rotational components $\boldsymbol{K}_o$ of the virtual stiffness matrix to 0 and hence achieving a "force-follow" behavior in the rotational directions.

5.3.4. Experimental Verification

Exemplary operated mechanisms are a cupboard door, a drawer and a microwave door providing linear (drawer) as well as arced constraint trajectories with different hinge locations. In addition, the cupboard door and the microwave door have the property that they require a certain break-away force ($20N$ and $15N$, respectively) before they start moving without any noteworthy resistance. The robot is always placed roughly in front of the device. The initial movement direction is set to $\boldsymbol{d}_{p_0} = (0, \sin(-\frac{\pi}{4}), \sin(-\frac{\pi}{4}))^T$ resulting in a large initial error angle of approximately 45°. The desired manipulation velocity ν is randomly varied between $0.01\frac{\text{m}}{\text{s}}$ and $0.05\frac{\text{m}}{\text{s}}$, the moving average window for the SMA is chosen out of the interval $N \in [300, 1000]$. With a sample frequency of $1kHz$, this results in a time window between 0.3 and 1 second. The admittance parameters $\boldsymbol{K}, \boldsymbol{D}$ and $\boldsymbol{M}$ are chosen heuristically, so that no contact instability occurred. Under these conditions all of the performed tests (20 for each device) were successful, see Fig. 5.8 for snapshots, i.e. a superior success rate of 100% is observed. In the following,

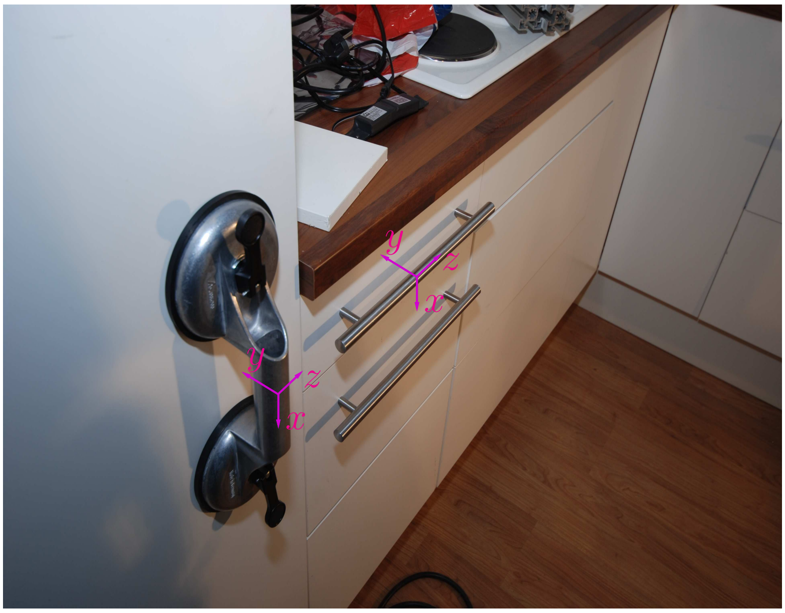


Figure 5.4. Experimental setup. The base coordinate system is plotted on two potential contact points to visualize the true direction of motion if the robot stands right in front of the devices.

the general performance and effects of some of the parameters of the proposed strategy are examined.

The influence of the SMA filtering can be observed in Fig. 5.5 where estimation process over time for two different time windows is shown for the opening of the cupboard door. Note that the estimation converges quickly to the true direction motion. As expected, a larger window leads to a smoother direction estimation. However, a larger phase lag is also observed, which potentially may result in increased contact forces especially at high velocities and/or mechanisms with small curvature radius.

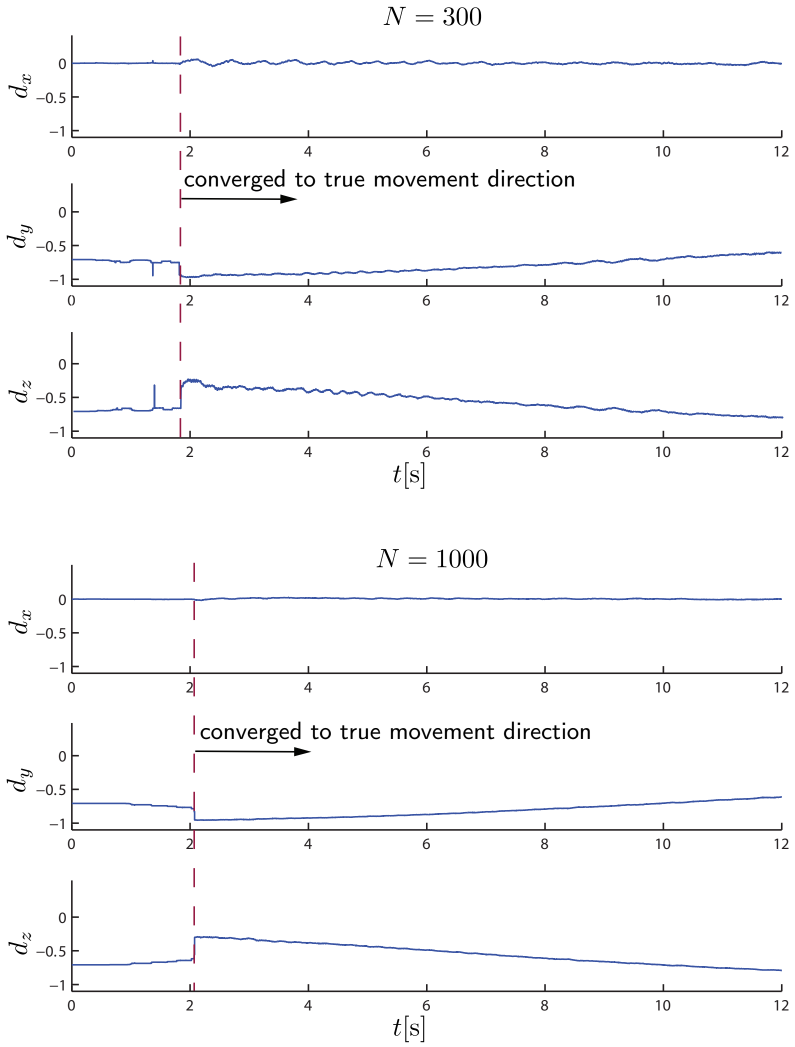


Figure 5.5. Effect of SMA filter order N for opening of a cupboard door: Improved noise reduction, but higher phase lag for larger N.

In order to validate the proposed projection approach, the opening of the drawer with the initial resisting force is performed. The initial movement direction $\boldsymbol{d}_{p_0}$ is again set to $(0, \sin(-\frac{\pi}{4}), \sin(-\frac{\pi}{4}))^T$. This results in an increase of $\boldsymbol{f}_{\parallel} = (0, f_y, 0)^T$ and $\boldsymbol{f}_{\perp} = (0, 0, f_z)^T$ with approximately the same slope as can be observed from Fig. 5.6. When the breakaway force is reached, $\boldsymbol{f}_{\parallel}$ vanishes quickly while $\boldsymbol{f}_{\perp}$ still remains even though the direction estimation has converged to the true movement direction $\boldsymbol{d}$. As a result, the contact force applied in y-direction remains nearly 20N. The projection in (5.6) leads to disappearance of the persistent contact forces, as shown in Fig. 5.7.

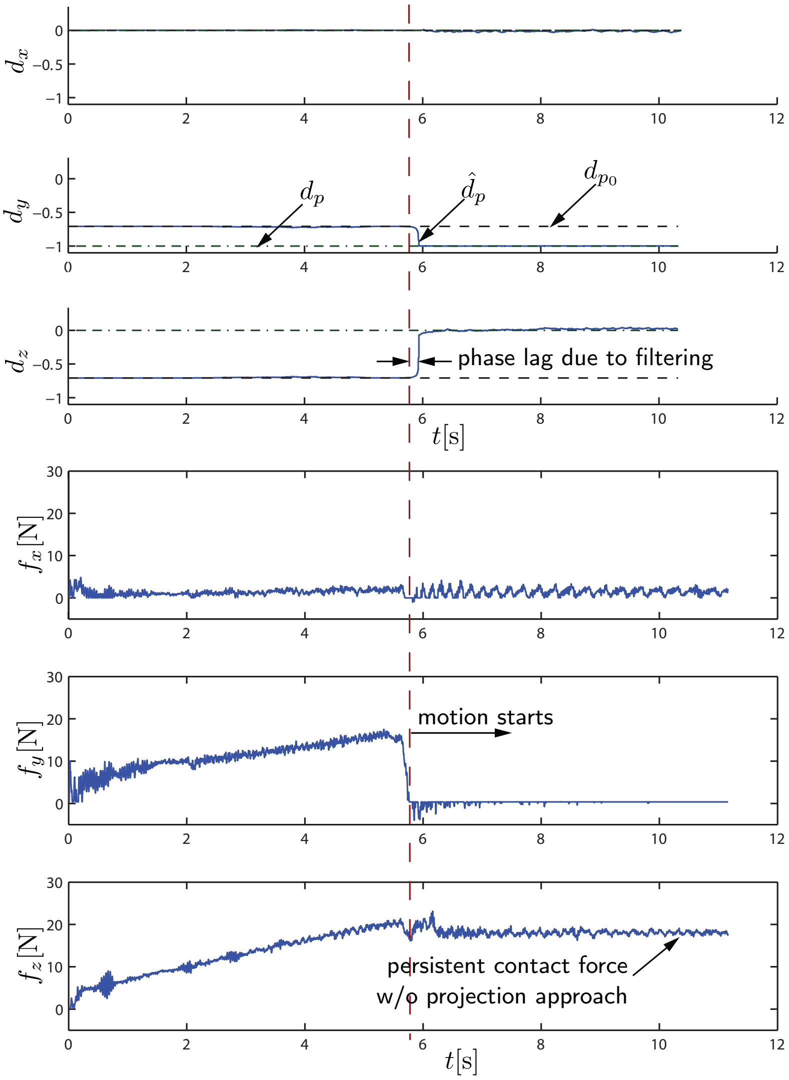


Figure 5.6. Direction estimation and contact force for opening drawer with breakaway force and initialization error of approx. 45° without projection approach.

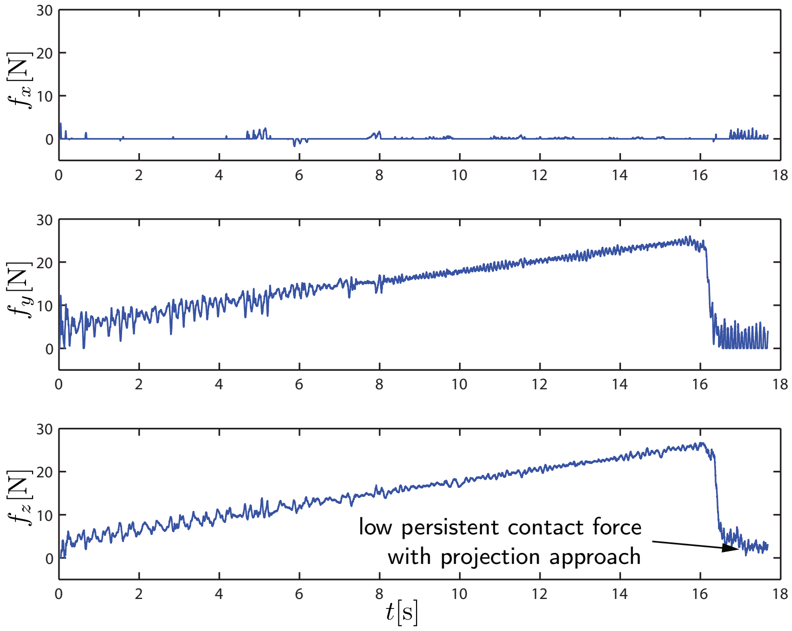


FIGURE 5.7. Contact force for opening drawer with breakaway force and initialization error of approx. 45° with projection approach.

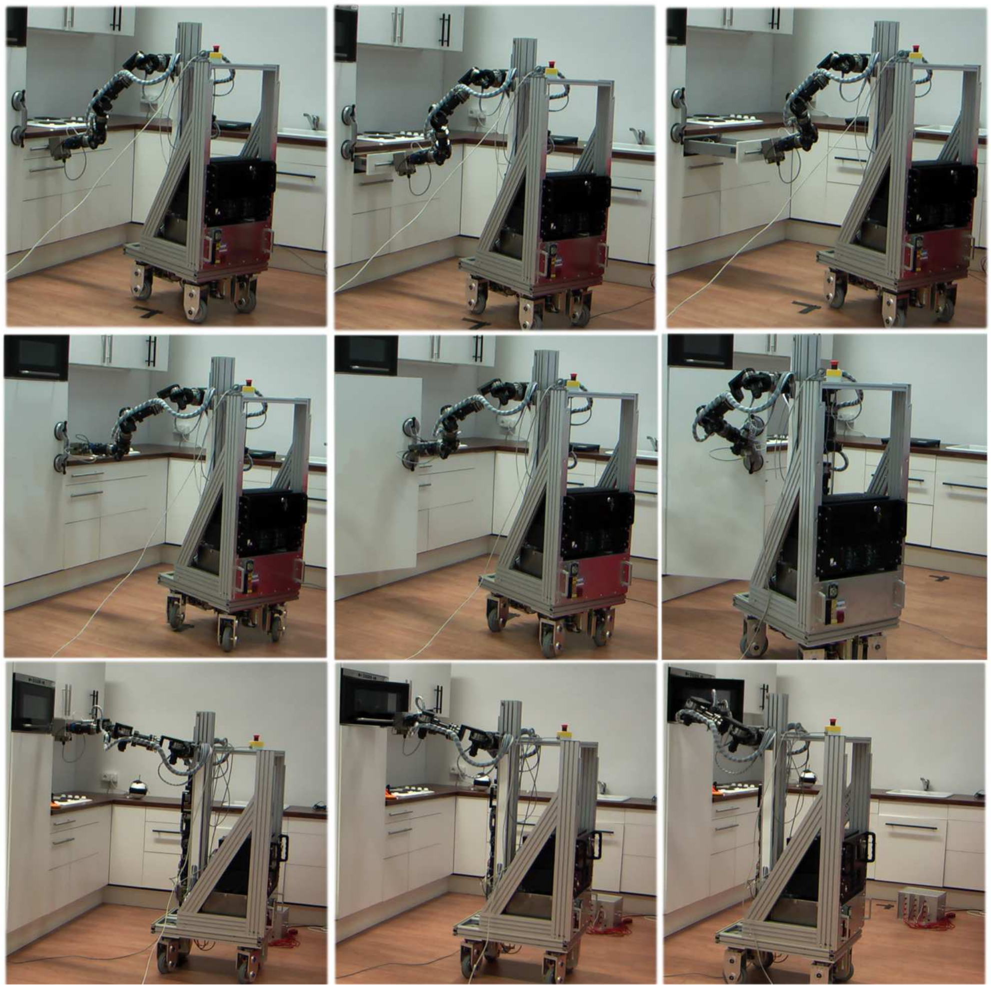


Figure 5.8. Mobile manipulator operating drawer, cupboard and microwave door using the proposed control strategy without change of parameters.

5.4. Extension to Joint-Level Compliance

The next step was to realize the constraint manipulation approach on a manipulator with compliant joints [67]. Applying the previously presented strategy directly, does not work in a satisfying way for joint-level compliance due to the misalignment of Cartesian position deviation $\boldsymbol{p}_v - \boldsymbol{p}$ and applied interaction forces $\boldsymbol{f}$ (see Fig. 2.8). The actual misalignment depends on the manipulator configuration, its kinematics and the joint controller gains, respectively virtual joint stiffness $\boldsymbol{K}$. For the kinematics of a KUKA LBR-IV (see appendix B), this misalignment can be easily larger than 50° for equal stiffness in every joint, even for descent values (> 0.1) of the manipulability μ (see equation (2.7)). This misalignment is also often neglected by many authors working with compliant joint controllers, e.g. [48].

5.4.1. Joint Space Exploration and Canceling Erroneous Forces

Having a discrete controller, the set-point is selected incrementally in order to accumulate a static force along $\hat{\boldsymbol{d}}_p$. As end-effector orientation is assumed locally unconstrained, the desired, normalized wrench direction is $[\hat{\boldsymbol{d}}_p \quad \boldsymbol{0}]^T$. Due to (2.12) the associated joint space position increment per time step, leading to the desired force increment $\kappa\hat{\boldsymbol{d}}_p\Delta T$, is

$$\boldsymbol{\delta q}_{\text{exp}} = \boldsymbol{K}_P^{-1}\boldsymbol{J}^T \begin{pmatrix} \kappa\hat{\boldsymbol{d}}_p \\ \boldsymbol{0} \end{pmatrix} \Delta T, \tag{5.7}$$

where the manipulation velocity is now parametrized by κ, denoting the desired force slope in $\frac{N}{s}$. From (5.7), the estimated direction in joint space is obtained with

$$\hat{\boldsymbol{d}}_q = \text{norm}(\boldsymbol{\delta q}_{\text{exp}}) \tag{5.8}$$

and compute the associated $n \times n$ projection matrix $\hat{\boldsymbol{P}}_q = \hat{\boldsymbol{d}}_q\hat{\boldsymbol{d}}_q^T$.

The set-point at the discrete time step $t[i+1]$ is finally obtained with

$$\boldsymbol{q}_v[i+1] = \boldsymbol{q}[i] + \hat{\boldsymbol{P}}_q(\boldsymbol{q}_v[i] - \boldsymbol{q}[i]) + \boldsymbol{\delta q}_{\text{exp}}[i], \tag{5.9}$$

where $\hat{\boldsymbol{P}}_q(\boldsymbol{q}_v[i]-\boldsymbol{q}[i])$ maps the applied static torques on $\hat{\boldsymbol{d}}_q$, effectively canceling out $\boldsymbol{f}_\perp$. This term is of major importance, since $\hat{\boldsymbol{d}}_q$ changes permanently with proceeding estimation of $\hat{\boldsymbol{d}}_p$, but also with changing manipulator configuration, due to the dependency

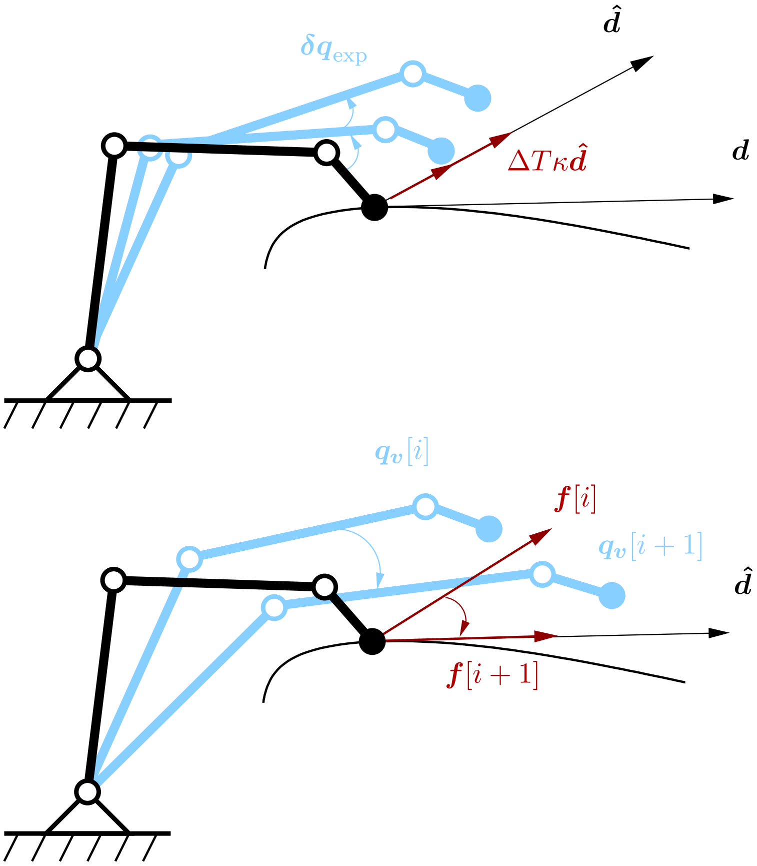


Figure 5.9. Constrained manipulation strategy for joint-level compliance. A joint space instantaneous command is executed to obtain accumulating interaction forces along the estimated direction of motion $\hat{\boldsymbol{d}}$. The full configuration is projected so that the interaction forces are always aligned with $\hat{\boldsymbol{d}}$.

of (5.7) on $\boldsymbol{J}(\boldsymbol{q})$.

5.4.2. Orientation Regulation

Due to the joint-level compliance, end-effector forces and rotational motion are not independent as for the admittance controller. The straight forward approach of adding a separate differential controller

$$\boldsymbol{\delta q}_o = \boldsymbol{J}_v^+ \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\omega}_d \end{pmatrix} \Delta T \tag{5.10}$$

to (5.9) leads still to good results, where $\boldsymbol{\omega}_d$ is the desired rotational velocity. The associated set-point generation rule is then

$$\boldsymbol{q}_v[i+1] = \boldsymbol{q}[i] + \hat{\boldsymbol{P}}_q(\boldsymbol{q}_v[i] - \boldsymbol{q}[i]) + \boldsymbol{\delta q}_{\mathrm{exp}}[i] + \boldsymbol{\delta q}_o. \tag{5.11}$$

The explanation for the good performance is that the projection $\hat{\boldsymbol{P}}_q(\boldsymbol{q}_v[i] - \boldsymbol{q}[i])$ assures that the applied static interaction forces are aligned with $\boldsymbol{d}_p$ and the applied moments are 0 in every control cycle. So the components of $\boldsymbol{\delta q}_o[i]$ which interfere with these conditions, are canceled in the next control cycle.

Assuming a Fixed Grasp With a fixed grasp, the update rule (5.9) leads to vanishing interaction moments and implements the same force-follow behavior as in Sec. 5.3, so that no extra orientation controller is required.

5.4.3. Experimental Verification

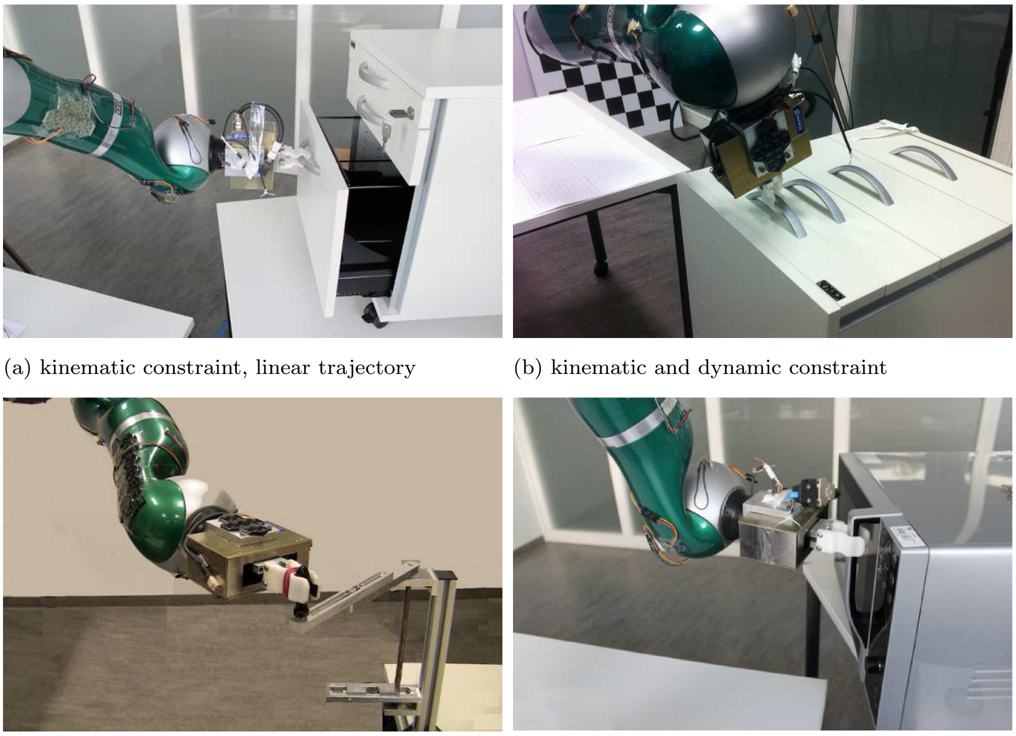
All graphs in this section show the estimation error angle ϵ between $\boldsymbol{d}_p$ and $\hat{\boldsymbol{d}}_p$ in black, the norm of the applied erroneous static forces $||\boldsymbol{f}_{s_\perp}||$ in red and the norm of the measured erroneous interaction forces in dashed magenta.

The control frequency was $500Hz$ and the stiffness was set to $\boldsymbol{K} = 40\boldsymbol{I}_7 Nm/\mathrm{rad}$ to achieve a high level of compliance. The order of the SMA was $N = 500$ and the dead-band threshold is $v_{\mathrm{th}} = 0.01m/s$. The force exploration rate κ and the corresponding exploration velocity ν were set to $9N/s$ and $0.05m/s$ to provide similar performance in terms of execution time for the operated mechanism. The parameters were chosen heuristically, to obtain satisfactory results in terms of execution time, stability and robustness.

The first set of experiments is a comparison between the approach presented here and the Cartesian trajectory-following strategy from Sec. 5.3 implemented on the LBR, running a joint space impedance controller. The task was to pull up a $1.2kg$ mass for $0.3m$. The motion of the mass was restricted to a linear trajectory. This setup was realized by a drawer in a vertical configuration (see Fig. 5.10b). The graphs in Fig. 5.11 show the data for the new approach and the trajectory-following strategy, each with different initial errors. It can be clearly seen, that for the trajectory-following strategy, where interaction forces are not considered, erroneous forces remain, even for very small estimation errors. This is due to the misalignment of the Cartesian position deviation and the applied static forces for joint space compliance. When applying (5.9) for set-point generation, erroneous forces build up for large initial estimation errors, until the mechanism starts moving and a descent estimation is obtained. After this, the erroneous forces vanish rapidly. Note also, that forces build up linearly for the proposed method according to κ, while in the Cartesian trajectory tracking approach ν has no relation to the slope of the forces.

For the measurements in Fig. 5.12, the drawer is in a horizontal configuration (see Fig. 5.10a), imposing a nearly pure kinematic constraint, with only small forces required. Also, in this case the new approach performs better in terms of applied erroneous forces, even though not as clearly as in the dynamically constrained case.

To test the new approach also for circular trajectories and evaluate its robustness, an experiment in turning a crank with varying radius and κ is performed (see Fig. 5.10c). As can be seen from the left graphs in Fig. 5.13, the new approach was successful, even for very small radii. However, one observes increasing estimation errors when shorten the radius. From the right graphs in Fig. 5.13, which show data for turning a crank with radius $17cm$ and varying the values for κ, it can be seen, how increased exploration speed leads to larger estimation errors and erroneous forces. This comes from the delay, introduced by the filter (5.4), which was also noticed in Sec. 5.3 and highlights the major



(a) kinematic constraint, linear trajectory

(b) kinematic and dynamic constraint

(c) circular trajectory

(d) switching of required interaction forces

Figure 5.10. Constraint manipulation tasks in unstructured environment.

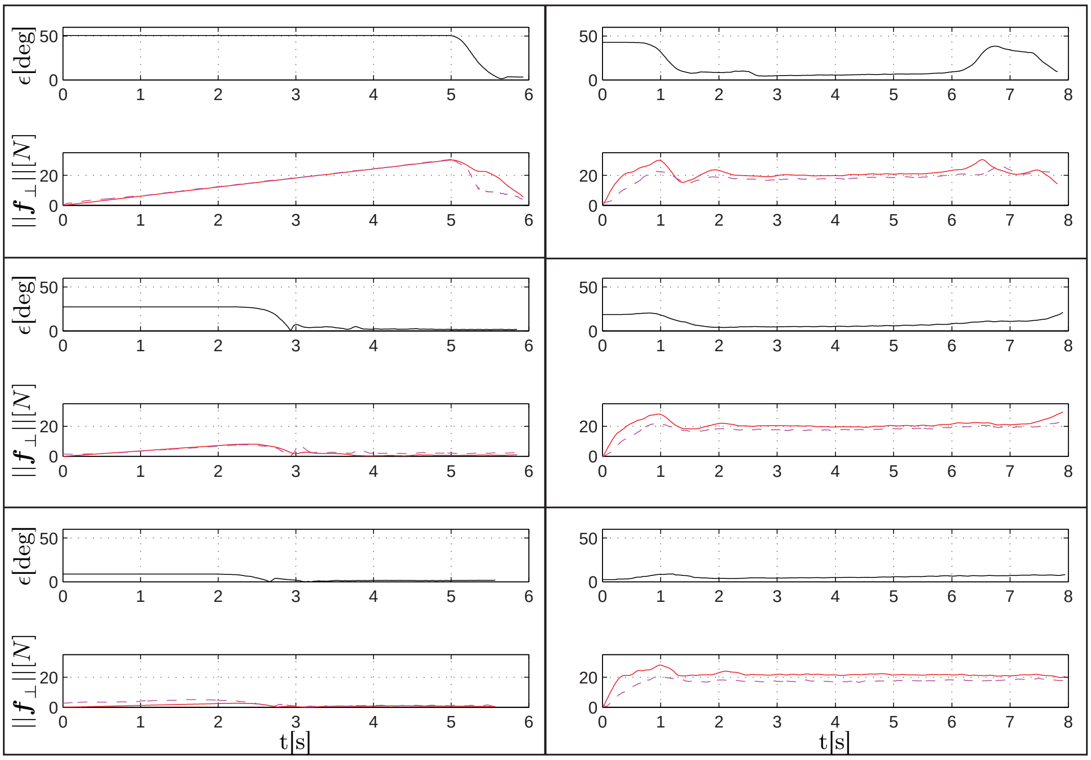


Figure 5.11. Pulling up a 1.2kg mass with different initial errors, using the new approach (left) and a Cartesian trajectory-following approach (right)

disadvantage of the simple filter-based estimation method.

The new approach also proved its robustness when manually disturbing the manipulation procedure or manipulating devices with changing required forces, e.g. microwave door in Fig. 5.10d.

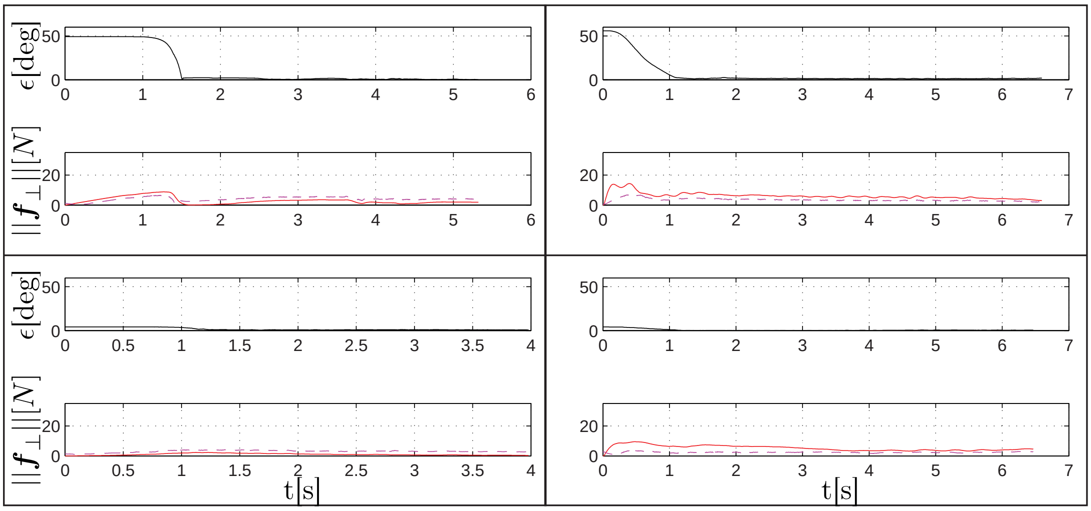


FIGURE 5.12. Opening a drawer (pure kinematic constraint) using the new approach (left) and a Cartesian trajectory-following approach (right)

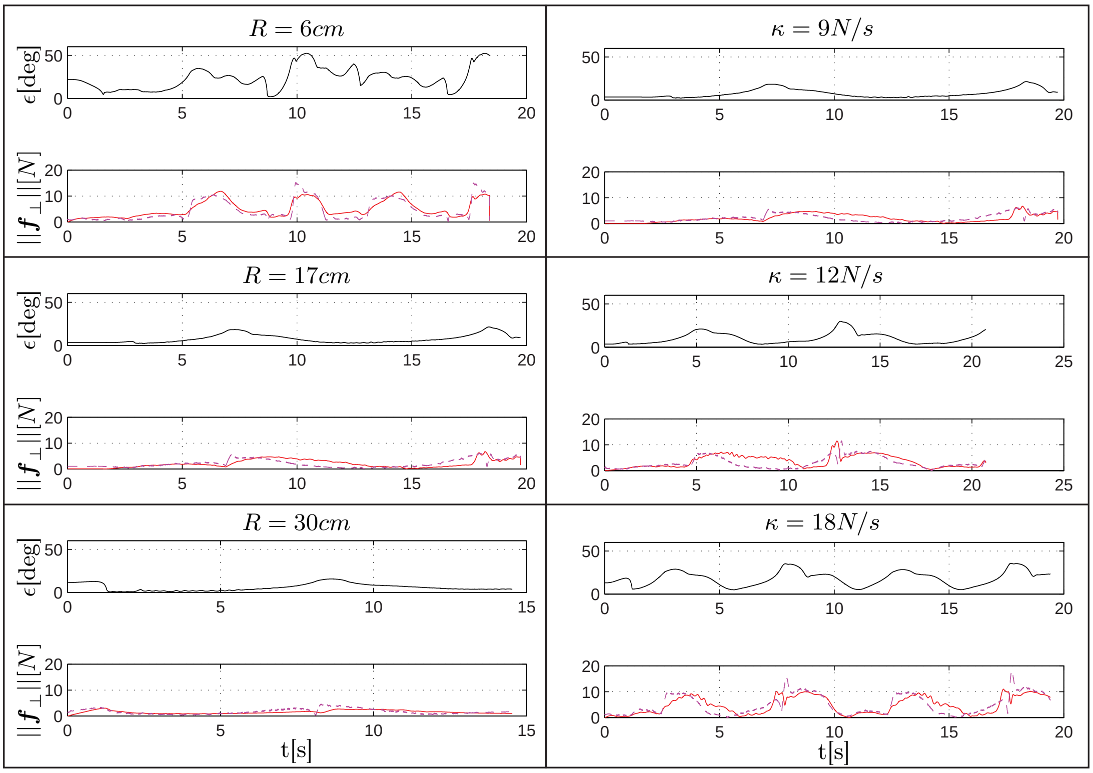


FIGURE 5.13. Left: Turning a crank with varying radius R; Right: varying exploration rate κ for $R = 17cm$

5.5. Using the New IFC Task Specification Scheme

While the solution proposed in 5.4 leads to satisfactory results, the need for a cleaner separation of force and orientation control and extension to a more general usage of a joint space IFC. In this section, the constraint manipulation task is formulated as a set of subtasks within the specification scheme, developed in Chapters 3 and 4.

5.5.1. Constraint Exploration and Canceling Erroneous Forces

The exploration regulator (5.9) has to be reformulated as a force task. Hence, the primary task type is end-effector wrench, which has to be formulated to keep the exploratory nature of (5.9). The applied force $\boldsymbol{f}_s$ needs to be projected on $\hat{\boldsymbol{d}}_{\boldsymbol{p}}$ so that the desired task variable is

$$\boldsymbol{\alpha}_{1_d} = \hat{\boldsymbol{P}}_p \boldsymbol{f}_s \tag{5.12}$$

Also, in this application there is no specific interaction force but instead the environmental constraint should be explored in a certain direction. Therefore, to make sure the applied forces level off at the required value, the velocity compensation term in the force task has to be canceled[2].

$$\dot{\boldsymbol{\alpha}}_{1_{\mathrm{ff}}} = \kappa \hat{\boldsymbol{d}}_{\boldsymbol{p}} - \boldsymbol{S}_2^T \boldsymbol{J}^{T+} \boldsymbol{K} \dot{\boldsymbol{q}} \tag{5.13}$$

The associated subspace matrix isolates the force components from the wrench:

$$\boldsymbol{S}_1 = \begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0}_3 \end{bmatrix} \tag{5.14}$$

5.5.2. Orientation Regulation and Remaining Degrees of Freedom

Having the constraint exploration on the top-level, regulation of the orientation can be treated in any desired way, since it will be executed in the nullspace of the exploration task. It is demonstrated how the developed task specification scheme can be used to easily adapt to different conditions or a priori knowledge.

[2] Canceling the velocity compensation term is just done here for formal correctness. In the implementation the velocity compensation can be switched on and off with a flag variable.

Fixed Orientation If the end-effector is supposed to hold its initial orientation $\boldsymbol{o}_{\text{init}}$, the second task specification would be

$$\boldsymbol{S}_2 = \begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix} \tag{5.15}$$

$$\boldsymbol{\alpha}_{2_d} = \boldsymbol{o}_{\text{init}}, \tag{5.16}$$

with task type Cartesian end-effector pose.

Align End-Effector Axis Usually the end-effector orientation is supposed to adjust itself to the constrained trajectory, e.g. when opening a door, the z-axis of the end-effector ($\boldsymbol{z}_{ee}$) should be aligned with the movement direction. The end-effector and base frame of the regarded setup is depicted in Fig. 5.15. Assuming the constraint does not impose a screw motion on the end-effector (rotation around $\boldsymbol{z}_{ee}$), the orientation regulation task can be specified to align $\boldsymbol{z}_{ee}$ with the current estimation $\hat{\boldsymbol{d}}_p$ using two subtasks:

$$\boldsymbol{S}_2 = \text{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee})[0\ 0\ 0\ 0\ 0\ 1]^T \tag{5.17}$$

$$\boldsymbol{\alpha}_{2_d} = \boldsymbol{S}_2^T \boldsymbol{x}_{\text{init}} \tag{5.18}$$

$$\boldsymbol{S}_3 = \text{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee}) \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.19}$$

$$\dot{\boldsymbol{\alpha}}_{\text{ff}_3} = k_o \boldsymbol{S}_3^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\boldsymbol{z}_{ee} \times \hat{\boldsymbol{d}}_p \end{bmatrix} \tag{5.20}$$

with $k_o > 0$ being a constant gain. $\text{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee})$ transforms the rotational components of a Cartesian subspace matrix or vector from end-effector to base coordinates. Here, task

2 keeps the orientation around $\boldsymbol{z}_{ee}$ constant, while task 3 aligns $\boldsymbol{z}_{ee}$ with $\hat{\boldsymbol{d}}_p$, assuming a pulling task[3]. For convenience, both subtasks are formulated in base coordinates and then transformed to the target system using $\boldsymbol{S}_{2/3}^T$. The two in principle independent orientation tasks are not combined in one single task as the constant orientation around $\boldsymbol{z}_{ee}$ is more crucial than the orientation around $\boldsymbol{x}_{ee}$ and $\boldsymbol{y}_{ee}$ due to the shape of the end-effector[4] and has higher priority in case the orientation subtask can not be fulfilled completely.

Release End-Effector Axis Assuming a caging grasp, another possibility for an orientation regulator can be defined, which would permit rotations around $\boldsymbol{y}_{ee}$. This direction could than be used by lower priority tasks. This orientation task would be specified with

$$\begin{aligned}\boldsymbol{S}_2 &= \operatorname{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee})[0\ 0\ 0\ 0\ 0\ 1]^T \\ \boldsymbol{\alpha}_{2_d} &= \boldsymbol{S}_2^T \boldsymbol{x}_{\text{init}}\end{aligned}$$

$$\begin{aligned}\boldsymbol{S}_3 &= \operatorname{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee})[0\ 0\ 0\ 1\ 0\ 0]^T \\ \dot{\boldsymbol{\alpha}}_{3_d} &= \boldsymbol{\Lambda}_3 \boldsymbol{S}_3^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\boldsymbol{z}_{ee} \times \boldsymbol{d} \end{bmatrix}\end{aligned}$$

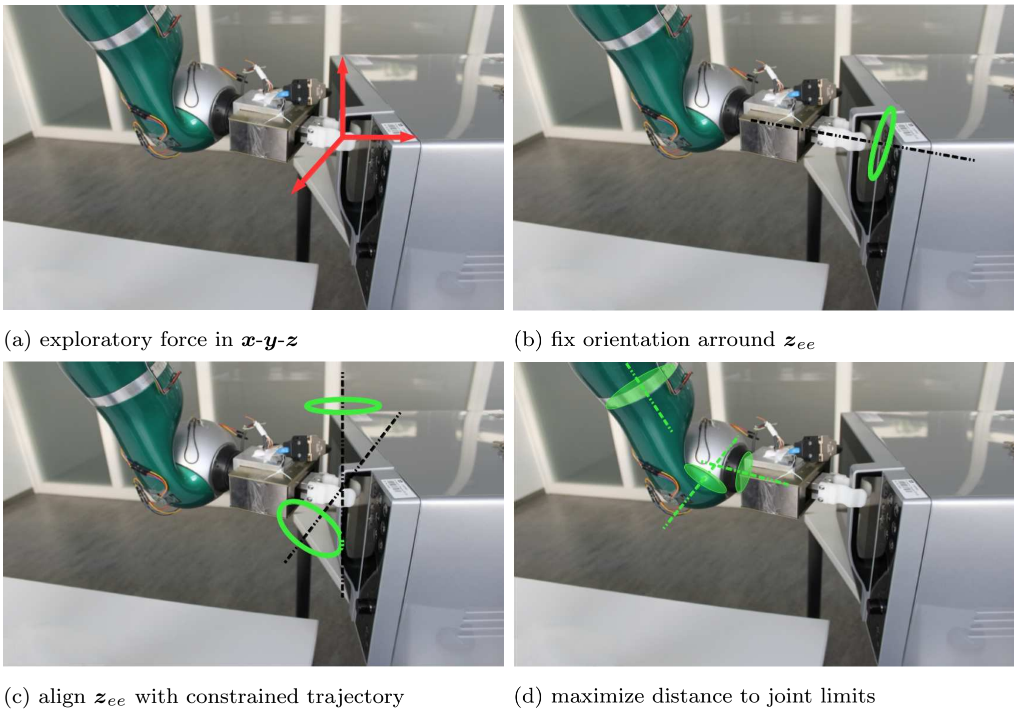
what shifts the rotational degree of freedom around $\boldsymbol{y}_{ee}$ to the nullspace of the manipulator with respect to that task.

Remaining Degrees of Freedom The rest of the manipulators degrees of freedom can be used in order to fulfill any standard manipulation nullspace task. For example, keeping the joints away from their limits (task type: joint position) or minimizing joint torques (task type: joint torques).

The exemplary task description for the fixed orientation case can be found in table 5.1 and a visualization of the subtasks can be seen in Fig. 5.14

[3] for pushing, $-\hat{\boldsymbol{d}}_p$ has to be used

[4] In real world applications the used gripper has still some freedom around $\boldsymbol{x}_{ee}$ and $\boldsymbol{y}_{ee}$ due to slippage. A screwing motion (rotation around $\boldsymbol{z}_{ee}$) however would directly induce moments at the end-effector and could lead to its damage.



(a) exploratory force in $\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{z}$

(b) fix orientation arround $\boldsymbol{z}_{ee}$

(c) align $\boldsymbol{z}_{ee}$ with constrained trajectory

(d) maximize distance to joint limits

FIGURE 5.14. Breaking down the constrained manipulation task into 4 subtasks.

5.5.3. Experimental Verification

The task is again to lift a $1.2kg$ mass to a height of $0.3m$, constrained to a linear trajectory (pulling a drawer in vertical direction, see Fig. 5.15 for the definition of base and end-effector frames). To compare the previous and the new approach independent from the constraint estimation, a constant ($\hat{\boldsymbol{d}}_p = (-1,\ 0,\ 0)^T$) was induced, which corresponds basically to the real direction of motion. However, erroneous forces still appear, especially for faster motions. This is due to unconsidered dynamic effects in combination with a not perfectly rigid constraint (slippage of grasp, play in the drawer mechanism), which lead to a deviation of the end-effector from the straight line.

For both approaches, the task is executed nine times with varying stiffness $\boldsymbol{K}$ and varying exploration rate κ. To quantify the increased performance when using the task specification framework compared to the previous approach, the *root mean square* (RMS) of the magnitude of the erroneous forces $||\boldsymbol{f}_\perp||$ are compared against each other. The effects of different stiffness and exploration rate parameters are discussed in Figs. 5.16 and 5.17. Both show $||\boldsymbol{f}_\perp||$ for two samples of each approach. With the orientation task no longer interfering with the force task, the overall RMS of $||\boldsymbol{f}_\perp||$ for the experiments with fixed

Table 5.1. Set of subtasks for constrained manipulation

task	type	$\boldsymbol{\alpha}_d$	$\dot{\boldsymbol{\alpha}}_{\text{ff}}$	$\boldsymbol{S}$
1	wrench	$\hat{\boldsymbol{P}}_p \boldsymbol{f}_s$	$\kappa \hat{\boldsymbol{d}}_{\boldsymbol{p}} - \boldsymbol{S}_2^T \boldsymbol{J}^{T+} \boldsymbol{K} \dot{\boldsymbol{q}}$	$\boldsymbol{S}_2 = \begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0}_3 \end{bmatrix}$
2	cart. pose	$\boldsymbol{o}_{\text{init}}$	$\boldsymbol{0}$	$\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$
3	joint position	$\boldsymbol{0}$	$\boldsymbol{0}$	$\boldsymbol{I}_7$

estimation $\hat{\boldsymbol{d}}_p$, was reduced from $0.58N$ to $0.17N$, which corresponds to a reduction of 70%.

For a more qualitative comparison, six trials using the SMA (5.4) as constraint estimator are executed. As expected, an even higher performance gain was achieved, as the reduced erroneous forces lead to a decreased estimation error, what on its part again reduces the erroneous forces. The RMS was reduced from $11.17N$ to $1.54N$, what corresponds to a reduction of 86%.

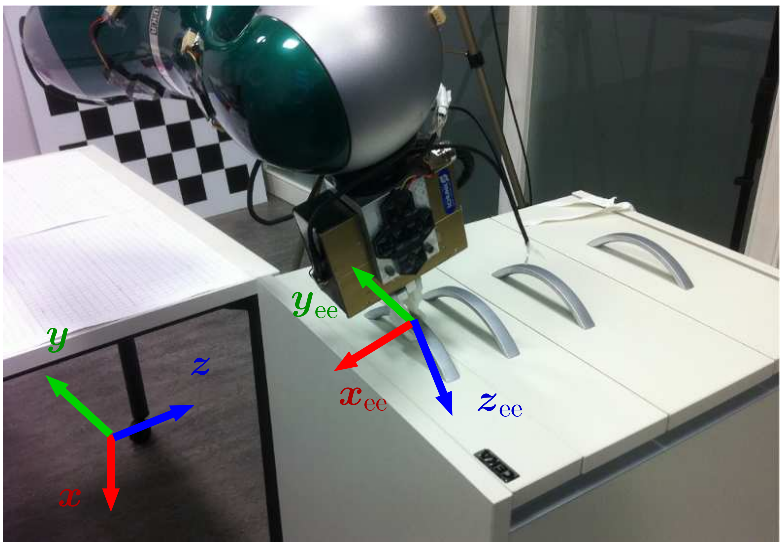

$\boldsymbol{y}_{\text{ee}}$
$\boldsymbol{y}$
$\boldsymbol{z}$
$\boldsymbol{x}_{\text{ee}}$
$\boldsymbol{z}_{\text{ee}}$
$\boldsymbol{x}$

Figure 5.15. The experimental setup with end-effector and base frame. The drawer has a mass of approximately 1.2 kg, hence requires active interaction forces for successful manipulation.

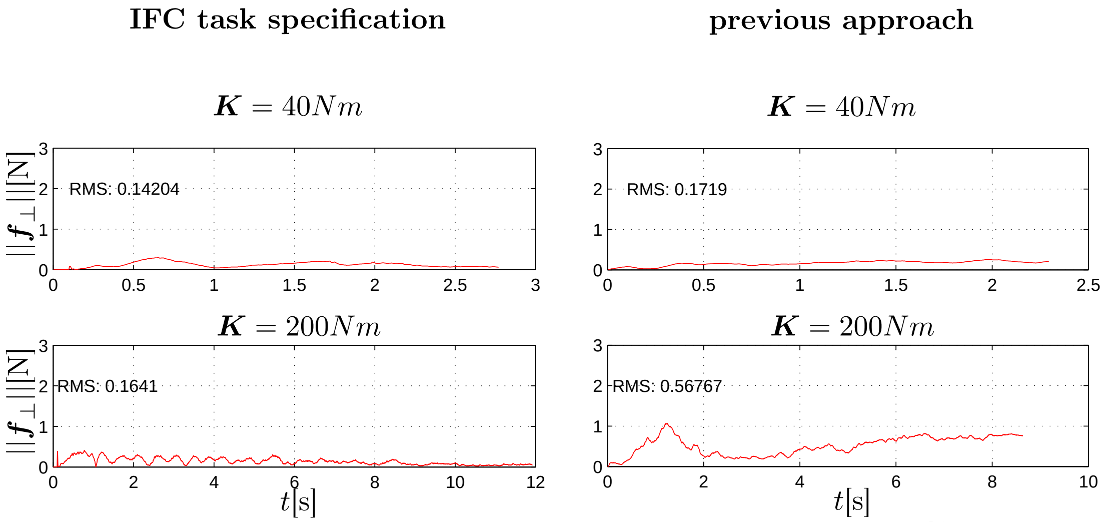


Figure 5.16. Pulling the drawer with fixed exploration speed $\kappa = 10\frac{N}{s}$ and varying stiffness $\boldsymbol{K}$. Left: utilizing the proposed task specification scheme for indirect force controlled robots. Right: previous approach with conflicting subtasks. For high compliance ($\boldsymbol{K} = 40Nm$), the two approaches perform almost equal, as the positioning task has only minor impact on the force task. With a stiffer manipulator ($\boldsymbol{K} = 200Nm$), the sensitivity of the force task to the positioning task is higher, what leads to increased erroneous forces when using the previous approach. The higher stiffness results also in a higher execution time, as the motion speed is reduced due to the higher sensitivity of the force task to $\dot{\boldsymbol{q}}_v$.

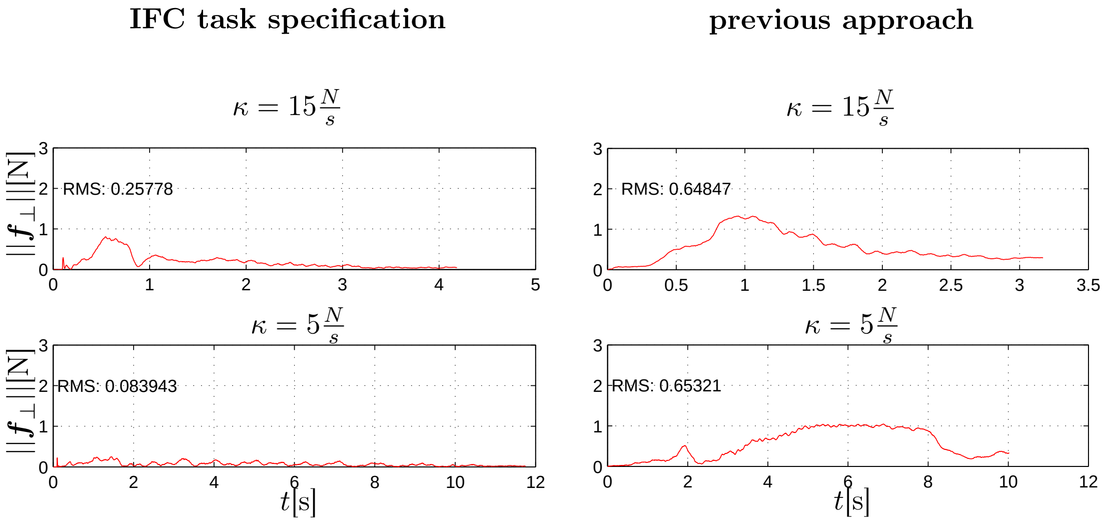


Figure 5.17. Pulling the drawer with fixed stiffness $\boldsymbol{K} = 100Nm$ and varying exploration rate κ. Left: utilizing the proposed task specification scheme for indirect force controlled robots. Right: previous approach with conflicting subtasks. The improved performance of the new approach is apparent for lower exploration rates ($\kappa = 5\frac{N}{s}$), as dynamic effects are almost not existent and the proposed task specification framework generates effectively no erroneous forces. With a higher exploration rate ($\kappa = 15\frac{N}{s}$) also the effective erroneous forces rise slightly, especially in the beginning, where acceleration is higher. For the previous approach, the induced erroneous forces due to conflicting subtasks are dominating and are not noticeably increased by a higher exploration rate.

5.6. Concluding Remarks

In this chapter, a novel manipulator control strategy for operating unknown constrained one–DoF mechanisms such as doors or drawers is proposed. Only few very general assumptions are made regarding the operated mechanism as well as the robotic manipulator. The first-order constraint parameter is continuously estimated and proper set-points for the underlying IFC are generated to explore the constraint trajectory along this direction and cancel erroneous forces. Besides the usage of a minimal model (constraint tangent), considering active interaction forces and dispense with special hardware devices (e.g. tactile sensors), the main advantage of the presented approach is, that not another control architecture is introduced but an application for the well studied IFC scheme is provided instead, which has proven its robustness for physical interaction tasks.

Three implementations were presented and discussed. The first is the original simple strategy for an admittance controlled manipulator. Then this strategy was generalized to robots with joint-level IFC's. Finally, the performance was improved by using the previously developed task specification scheme for indirect force controlled robots, i.e. the erroneous forces were significantly reduced by proper separation of the force and positioning task.

Chapter 6

Conclusion and Outlook

In this thesis, different ideas from robot control were combined to develop an additional layer between the task programming and the low level indirect force controlled robot. The assumptions on the actual robotic hardware are very general, hence the formulated methods are applicable on any IFC-like architecture or also SEA type manipulators, which represent a physical realization of an IFC. The resulting scheme allows the specification of a hierarchical set of equality and inequality tasks, consisting of force and positioning components on joint and Cartesian level. With this task specification interface, the application programmer is provided with an intuitive, yet powerful tool for task programming in a highly unstructured environment.

Based on simplifying assumptions, a set of task archetypes was formulated and a generalized task regulation scheme was derived. The resulting formalism made it possible to apply state of the art methods for multi-task programming and redundancy resolution in the context of IFC. A task can be composed of hierarchically ordered equality and inequality subtasks on joint and Cartesian level, which are provided by the application programmer. The set-points for the IFC are generated based on this high level task description as depicted in Fig. 6.1.

During the work for this thesis, a novel approach to operate highly constrained mechanisms was developed. Based on simple first-order estimation, the inherent compliance of the underlying IFC is exploited to explore the constraint without exact knowledge on its geometry or on the required interaction forces.

While the proposed approach on programming IFC robots is complete in itself, it leaves room for extensions and opens up new research directions.

A next step would be to implement the proposed scheme on a robot with physically

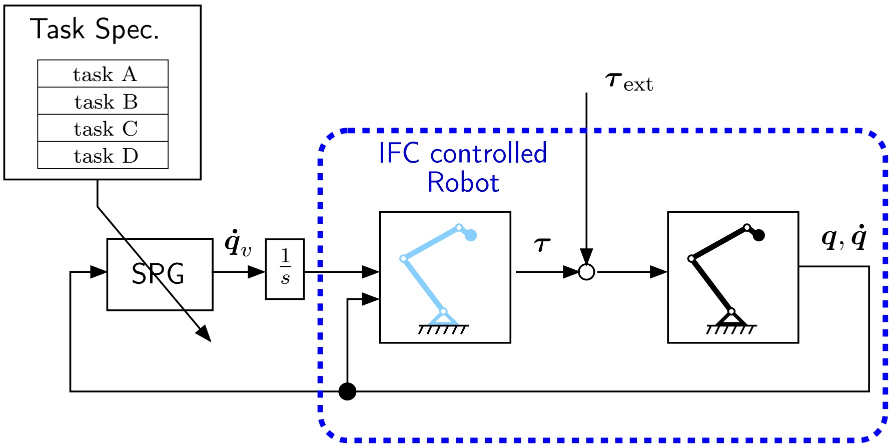


Figure 6.1. The IFC-robot block is considered a black box, which returns the robot state and takes set-point commands for the virtual robot as input. These set-points are provided by the SPG, depending on the specified hierarchical subtasks in the task specification block and the robot state.

compliant joints (SEA type robot). It is expected that the results are comparable to the IFC case, due to the physical equivalence of the problem.

To increase the practical relevance, it would be interesting to see if and how the proposed approach can be integrated in a planning scheme.

A case that was not considered in this thesis, but which has a high practical relevance, is the mixture of compliant and stiff joints in one kinematic structure. This applies to robots which have only partially compliant joints or also mobile manipulators, where the DOFs controlled by the mobile base can be considered stiff. Problems to solve are identification and handling of non-compliant (stiff) directions, which are now not identical to singular directions. In other words, the robot may apply forces in directions, which are not compliant.

Appendix A

Publications

- **Ewald Lutscher**, Emmanuel C. Dean-León and Gordon Cheng. *Hierarchical Force and Positioning Task Specification for Indirect Force Controlled Robots.* Submitted to IEEE Transactions on Robotics (conditionally accepted).

- **Ewald Lutscher** and Gordon Cheng. *Hierarchical Iinequality Task Specification for Indirect Force Controlled Robots Using Quadratic Programming.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Chicago, USA. October 14 - 18, 2014.

- **Ewald Lutscher** and Gordon Cheng. *Constrained Manipulation in Unstructured Environment Utilizing Hierarchical Task Specification for Indirect Force Controlled Robots.* IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China. May 31 - June 7, 2014.

- Karinne Ramirez Amaro, **Ewald Lutscher**, Andreas Holzbach and Gordon Cheng. *iCub@ICS-TUM: Semantic Reasoning, Constrained Manipulation and Humanoid Vision Enable on the iCub.* IEEE International Conference on Robotics and Automation (ICRA Workshop). Hong Kong, China. May 31 - June 7, 2014.

- **Ewald Lutscher** and Gordon Cheng. *A Practical Approach to Generalized Hierarchical Task Specification for Indirect Force Controlled Robots.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Tokyo, Japan. November 3 - 8, 2013.

- **Ewald Lutscher** and Gordon Cheng. *A Set-Point-Generator for Indirect-Force-Controlled Manipulators Operating Unknown Constrained Mechanisms.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vilamoura, Portugal. October 7 - 12, 2012.

- **Ewald Lutscher** and Gordon Cheng. *Operating Unknown Constrained Mechanisms with Compliant Robots.* 7th German Conference on Robotics (ROBOTIK). Munich, Germany. May 21 - 22, 2012.
- **Ewald Lutscher**, Martin Lawitzky, Gordon Cheng and Sandra Hirche. *A Control Strategy for Operating Unknown Constrained Mechanisms.* IEEE International Conference on Robotics and Automation (ICRA). Anchorage, USA. May 3 - 7, 2010.

Appendix B

Experimental Platforms

This appendix describes the robotic platforms used in the experiments for this thesis.

B.1. Kuka LBR-IV

The LBR comes with a sophisticated interface, which provides a variety of different control modes, i.e. joint position control, Cartesian-/joint-impedance control with tunable impedance parameters and direct torque input. Nevertheless, only the joint impedance control mode is used to maintain a high generalization among different platforms.

The controller is formulated as

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}) + \boldsymbol{d}_{fri} + \boldsymbol{\tau}_{fri} + \boldsymbol{\tau}_{dyn}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \tag{B.1}$$

with $\boldsymbol{d}_{fri}$ as a damping term depending on user defined damping parameters for each joint and $\boldsymbol{\tau}_{dyn}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ denoting the compensation for dynamical effects, including gravity. The optional additional torque $\boldsymbol{\tau}_{fri}$ is set to $\boldsymbol{0}$ for all the applications in this thesis. This controller mirrors very accurately a physical mass-spring-damper system on joint-level.

The default joint limits can be looked up in table B.1 and the kinematic structure is depicted in Fig. B.1.

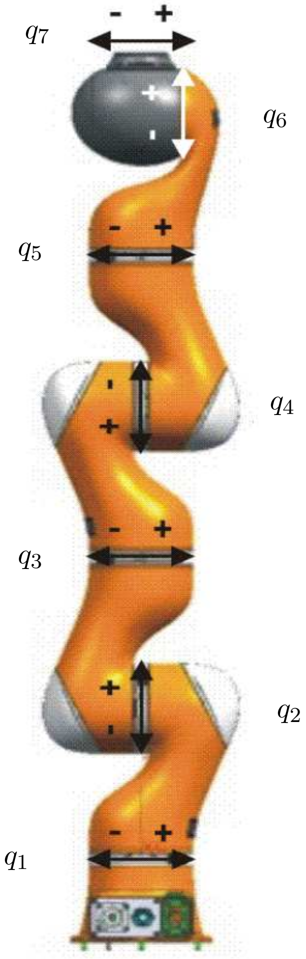


Figure B.1. Kinematic structure of the Kuka LBR-IV.

Table B.1. LBR – Default joint level limits

joint ID	$\boldsymbol{q}[deg]$	$\dot{\boldsymbol{q}}[\frac{deg}{s}]$	$\ddot{\boldsymbol{q}}[\frac{deg}{s^2}]$	$\boldsymbol{\tau}[Nm]$
1	$[-170, 170]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
2	$[-120, 120]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
3	$[-170, 170]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
4	$[-120, 120]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
5	$[-170, 170]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
6	$[-105, 105]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$
7	$[-170, 170]$	$[-20, 20]$	$[-500, 500]$	$[-50, 50]$

B.2. ACCREA Dual Arm Mobile Manipulator

This dual arm system was developed within the scope of a PhD thesis at the Technische Universität München [109]. It is built from commercially available components combined with aluminum/steel construction elements. The actuation torque is provided by DC-motors coupled with harmonic drive gears offering zero backlash. The motors are actuated by PWM-amplifiers supplying a control of the motor current at a bandwidth of 2.5 kHz. In order to permit force feedback control the devices are equipped with a six-axis *force-torque sensor* (FTS), each providing a bandwidth of 8 kHz at a comparatively low noise level. The joint angles are measured by digital MR-encoders with a resolution of 4096 counts per revolution, resulting in a comparatively high position resolution when multiplied with the corresponding gear ratio. The position based admittance controller from Sec. 2.3.2 is implemented. Fig. B.2 depicts the structure and the base frame coordinate system.

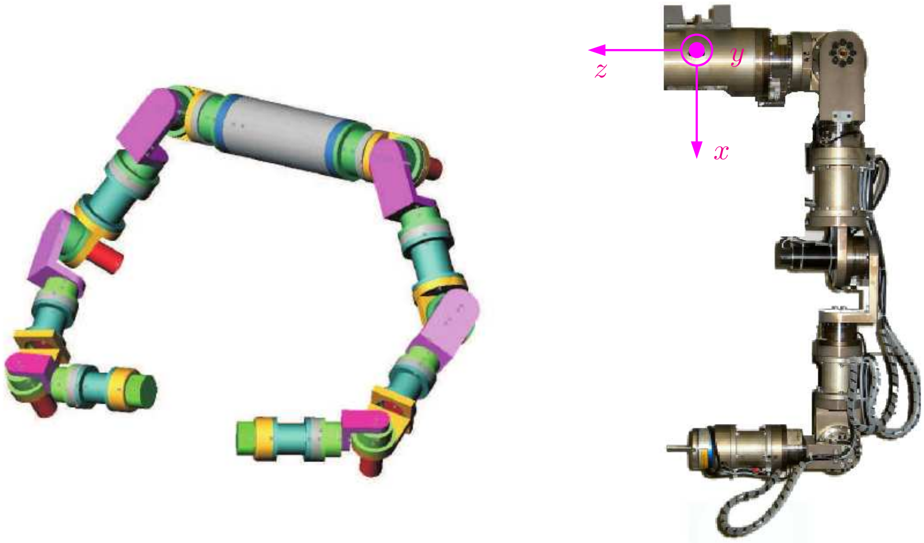


Figure B.2. Accrea Dual arm system. The right figure shows the left arm. All the planning of the arm movement is done within the depicted frame with origin between the two arms. Figures are taken from [109].

B.3. iCub Humanoid Robot

The iCub robot is a 94cm tall humanoid robot and was primarily designed to embody a cognitive system [97]. Therefore the focus was not on accurate position control and the first versions, having no force sensing, were very limited in executing contact tasks.

Adding force sensing in the shoulders and hips, together with proper interpretation of the joint currents, allowed to realize an active compliance controller in the form of an impedance-like interface [42]. This accepts either position or velocity commands, which are transformed to desired motor torques according to

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\text{offset}}, \tag{B.2}$$

where $\boldsymbol{\tau}_{\text{offset}}$ is an additional desired offset torque which can be set by the user. Fig. B.3 depicts the kinematic structure and the location of the FTS's.

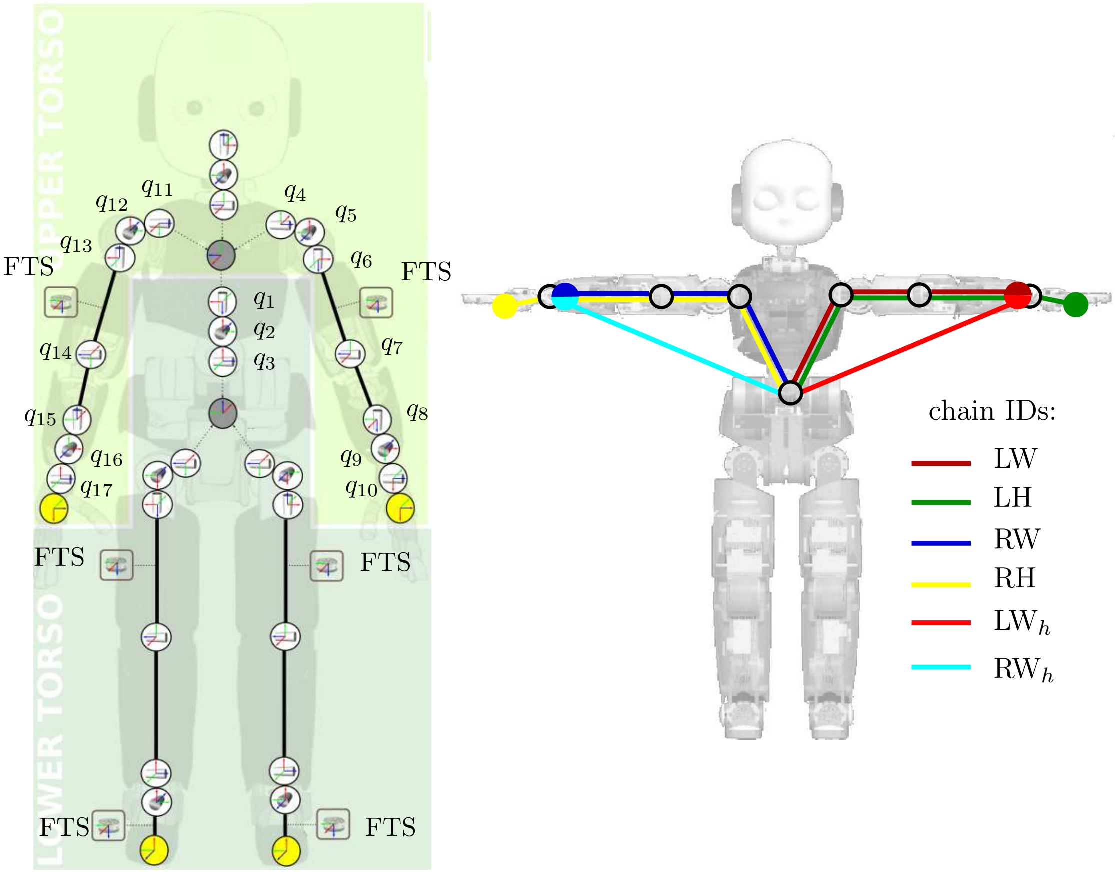


Figure B.3. *Left:* Kinematic structure of the iCub and location of the FTS's. *Right:*Kinematic trees, defined for interaction tasks. For each arm two end points are defined, one in the wrist and one in the hand. As the hand is not compliant, force tasks are executed at the wrist only.

Appendix C

Algorithms

This appendix summarizes the main methodologies presented in this thesis as algorithms for easy reimplementation.

Every algorithm is executed iteratively as part of a control loop to obtain a $\dot{\boldsymbol{q}}_v$, which serves as an input to the IFC. The following functions are defined:

- $\mathit{append}(\boldsymbol{x}, i)$ – adds element i to the vector $\boldsymbol{x}$
- $\mathit{clear}(\boldsymbol{x})$ – removes all the elements from the vector $\boldsymbol{x}$
- $\mathit{isempty}(\boldsymbol{x})$ – returns *true* if the vector $\boldsymbol{x}$ is empty, *false* otherwise
- $\boldsymbol{X}(:, \boldsymbol{i})$ – selects the rows of the Matrix $\boldsymbol{X}$, which are defined in the vector of indexes $\boldsymbol{i}$

SPG based on Hierarchical Nullspace Mapping is basically the application of the results of Chapter 3. $\boldsymbol{l}_q$ is a vector, which stores the indexes of the joints close to their limits. In line 7 the task Jacobian depending on the task type is set, while in line 10 the effect of the physical robot's joint velocities are compensated for, c.f. table 3.1. In line 8 the joints, which are close to their limits are clamped by setting the associated columns of the task Jacobian to zero. Line 16 calls algorithm C.2 to check for joints close to their limits.

checklimits checks if a joint is within the critical threshold Δl_{crit} and returns a vector of indexes of the critical joints.

SPG based on QP implements the method presented in Chapter 4 without the PSPM stabilizer. The augmented task Jacobian and QP limits are initialized with zeros in line 1. The general optimization variable bounds are defined in line 11. No limits are imposed on the slack variable. In lines 12 to 14 the parameters for a general QP are defined. After solving this problem (line 15), the augmented bounds and the augmented task Jacobian are adjusted for the next subtask (line 19).

PassiveVelLimShaping shapes the velocity limits to enforce passivity. Some high numerical values are assigned to the joint velocity bounds $\dot{q}_{p_m}$ and $\dot{q}_{p_M}$ in line 1 of algorithm C.4. Depending on the sign of Δq, either the upper or lower velocity limit is adjusted in line 4 respectively line 7. The original limits $\dot{q}_{v_m}$ and $\dot{q}_{v_M}$ are replaced with the limits due to the remaining energy if they are more restrictive in lines 9 to 14.

SPG based on QP with PSPM This algorithm is the extension of algorithm C.3 with the PSPM stabilizer. For every joint, the passivity condition is incorporated with algorithm C.4 and the velocity limits are adjusted if necessary in line 12. The energy reservoir is updated in line 22 and the ceiling operation is applied in lines 23 to 25.

Algorithm C.1 SPG based on Hierarchical Nullspace Mapping

```
*append*($\boldsymbol{l}_q$, 1)
**while not**(*isempty*($\boldsymbol{l}_q$)) **do**
    *clear*($\boldsymbol{l}_q$)
    $\dot{\boldsymbol{q}}_v \leftarrow \boldsymbol{0}$
    $\boldsymbol{N} \leftarrow \boldsymbol{I}_n$
    **for** $i \leftarrow 1$ *to* $k$ **do**
        $\boldsymbol{A} \leftarrow \boldsymbol{S}_i^T \boldsymbol{A}(\text{taskType}_i)$
        $\boldsymbol{A}(\ :\ , \boldsymbol{l}_q) \leftarrow \boldsymbol{0}$
        $\dot{\boldsymbol{\alpha}}_d \leftarrow \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\alpha}}_i + \dot{\boldsymbol{\alpha}}_{i_{ff}}$
        $\dot{\boldsymbol{\beta}}_d \leftarrow \dot{\boldsymbol{\alpha}}_d + \boldsymbol{\gamma}(\text{taskType}_i)$
        $\dot{\boldsymbol{\beta}}_d \leftarrow \dot{\boldsymbol{\beta}}_d - \boldsymbol{A}\dot{\boldsymbol{q}}_v$
        $\boldsymbol{A} \leftarrow \boldsymbol{A}\boldsymbol{N}$
        $\dot{\boldsymbol{q}}_v \leftarrow \dot{\boldsymbol{q}}_v + \boldsymbol{A}^+ \dot{\boldsymbol{\beta}}_d$
        $\boldsymbol{N} \leftarrow \boldsymbol{N} - \boldsymbol{A}^+ \boldsymbol{A}$
    **end for**
    $\boldsymbol{l}_q \leftarrow$ *checklimits*($\boldsymbol{q}_v, \dot{\boldsymbol{q}}_{\boldsymbol{v}}$)
**end while**
**return** $\dot{\boldsymbol{q}}_v$
```

Algorithm C.2 checklimits($\boldsymbol{q}_v, \dot{\boldsymbol{q}}_{\boldsymbol{v}}$)

```
*clear*($\boldsymbol{l}_{\boldsymbol{q}}$)
**for** $i \leftarrow 1$ *to* $n$ **do**
    **if** $q_{i_M} - q_{i_v} < \Delta l_{\text{crit}}$ **and** $\dot{q}_{i_v} > 0$ **then**
        *append*($\boldsymbol{l}_{\boldsymbol{q}}$, $i$)
    **end if**
    **if** $q_{i_v} - q_{i_m} < \Delta l_{\text{crit}}$ **and** $\dot{q}_{i_v} < 0$ **then**
        *append*($\boldsymbol{l}_{\boldsymbol{q}}$, $i$)
    **end if**
**end for**
**return** $\boldsymbol{l}_{\boldsymbol{q}}$
```

Algorithm C.3 SPG based on QP

```
$\bar{\boldsymbol{A}} \leftarrow \boldsymbol{0}_{1\times n},\ \bar{\boldsymbol{b}}_m \leftarrow 0,\ \bar{\boldsymbol{b}}_M \leftarrow 0$
**for** $i \leftarrow 1$ *to* $k$ **do**
    $m \leftarrow \dim(\boldsymbol{\alpha}_i)$
    $\boldsymbol{A} \leftarrow \boldsymbol{S}_i^T \boldsymbol{A}(\mathrm{taskType}_i)$
    $\dot{\boldsymbol{\alpha}}_m \leftarrow \boldsymbol{\Lambda}_i(\boldsymbol{\alpha}_{i_m} - \boldsymbol{\alpha}_i) + \dot{\boldsymbol{\alpha}}_{i_{m_{ff}}}$
    $\dot{\boldsymbol{\alpha}}_M \leftarrow \boldsymbol{\Lambda}_i(\boldsymbol{\alpha}_{i_M} - \boldsymbol{\alpha}_i) + \dot{\boldsymbol{\alpha}}_{i_{M_{ff}}}$
    $\dot{\boldsymbol{\alpha}}_m \leftarrow \dot{\boldsymbol{\alpha}}_m + \boldsymbol{\gamma}(\mathrm{taskType}_i)$
    $\dot{\boldsymbol{\alpha}}_M \leftarrow \dot{\boldsymbol{\alpha}}_M + \boldsymbol{\gamma}(\mathrm{taskType}_i)$
    $\dot{\boldsymbol{q}}_{v_m} \leftarrow \max\{\frac{\boldsymbol{q}_m - \boldsymbol{q}}{T},\ -\boldsymbol{v}_M,\ -\sqrt{2\boldsymbol{a}_M(\boldsymbol{q} - \boldsymbol{q}_m)}\}$
    $\dot{\boldsymbol{q}}_{v_M} \leftarrow \min\{\frac{\boldsymbol{q}_M - \boldsymbol{q}}{T},\ \boldsymbol{v}_M,\ \sqrt{2\boldsymbol{a}_M(\boldsymbol{q}_M - \boldsymbol{q})},\}$
    $\boldsymbol{w}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_{v_m} \\ -\infty \end{bmatrix}, \quad \boldsymbol{w}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_{v_M} \\ \infty \end{bmatrix}$
    $\boldsymbol{H} \leftarrow \begin{bmatrix} \rho \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_m \end{bmatrix}$
    $\boldsymbol{C} \leftarrow \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{I}_m \\ \bar{\boldsymbol{A}} & \boldsymbol{0} \end{bmatrix}$
    $\boldsymbol{b}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_m \\ \bar{\boldsymbol{b}}_m \end{bmatrix}, \quad \boldsymbol{b}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_M \\ \bar{\boldsymbol{b}}_M \end{bmatrix}$
    $\boldsymbol{w}^* \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_v^* \\ \boldsymbol{s}^* \end{bmatrix} \leftarrow \textit{solveQP}()$
    $\bar{\boldsymbol{b}}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_m + \boldsymbol{s}^* \\ \bar{\boldsymbol{b}}_m \end{bmatrix}, \quad \bar{\boldsymbol{b}}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_M + \boldsymbol{s}^* \\ \bar{\boldsymbol{b}}_M \end{bmatrix}, \quad \bar{\boldsymbol{A}} \leftarrow \begin{bmatrix} \boldsymbol{A} \\ \bar{\boldsymbol{A}} \end{bmatrix}$
**end for**
**return** $\dot{\boldsymbol{q}}_v^*$
```

Algorithm C.4 PassiveVelLimShaping

```
$\dot{q}_{p_m} \leftarrow -1e^{10}, \quad \dot{q}_{p_M} \leftarrow 1e^{10}$
$E_{D_{\min}} \leftarrow \frac{D}{\Delta T}(q_{\max}(N-1) - q_{\min}(N-1))^2$
if $\Delta q \geq 0$ then
    $\dot{q}_{p_M} \leftarrow \frac{1}{\Delta T}(\sqrt{\frac{2}{K}(E_r + E_{D_{\min}} + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q)$
end if
if $\Delta q \leq 0$ then
    $\dot{q}_{p_m} \leftarrow \frac{1}{\Delta T}(-\sqrt{\frac{2}{K}(E_r + E_{D_{\min}} + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q)$
end if
if $\dot{q}_{v_m} < \dot{q}_{p_m}$ then
    $\dot{q}_{v_m} \leftarrow \dot{q}_{p_m}$
end if
if $\dot{q}_{v_M} > \dot{q}_{p_M}$ then
    $\dot{q}_{v_M} \leftarrow \dot{q}_{p_M}$
end if
return $E_{D_{\min}}, \dot{q}_{v_m}, \dot{q}_{v_M}$
```

Algorithm C.5 SPG based on QP with PSPM

```
$\bar{\boldsymbol{A}} \leftarrow \boldsymbol{0}_{1\times n},\ \bar{\boldsymbol{b}}_m \leftarrow 0,\ \bar{\boldsymbol{b}}_M \leftarrow 0$
**for** $i \leftarrow 1$ *to* $k$ **do**
    $m \leftarrow \dim(\boldsymbol{\alpha}_i)$
    $\boldsymbol{A} \leftarrow \boldsymbol{S}_i^T \boldsymbol{A}(\text{taskType}_i)$
    $\dot{\boldsymbol{\alpha}}_m \leftarrow \boldsymbol{\Lambda}_i(\boldsymbol{\alpha}_{i_m} - \boldsymbol{\alpha}_i) + \dot{\boldsymbol{\alpha}}_{m_{i_{ff}}}$
    $\dot{\boldsymbol{\alpha}}_M \leftarrow \boldsymbol{\Lambda}_i(\boldsymbol{\alpha}_{i_M} - \boldsymbol{\alpha}_i) + \dot{\boldsymbol{\alpha}}_{M_{i_{ff}}}$
    $\dot{\boldsymbol{\alpha}}_m \leftarrow \dot{\boldsymbol{\alpha}}_m + \boldsymbol{\gamma}(\text{taskType}_i)$
    $\dot{\boldsymbol{\alpha}}_M \leftarrow \dot{\boldsymbol{\alpha}}_M + \boldsymbol{\gamma}(\text{taskType}_i)$
    $\dot{\boldsymbol{q}}_{v_m} \leftarrow \max\{\frac{\boldsymbol{q}_m - \boldsymbol{q}}{T},\ -\boldsymbol{v}_M,\ -\sqrt{2\boldsymbol{a}_M(\boldsymbol{q} - \boldsymbol{q}_m)}\}$
    $\dot{\boldsymbol{q}}_{v_M} \leftarrow \min\{\frac{\boldsymbol{q}_M - \boldsymbol{q}}{T},\ \boldsymbol{v}_M,\ \sqrt{2\boldsymbol{a}_M(\boldsymbol{q}_M - \boldsymbol{q})},\}$
    **for** $j \leftarrow 0$ *to* $n$ **do**
        $[E_{D_{\min j}}, \dot{q}_{j_{v_m}}, \dot{q}_{j_{v_M}}] \leftarrow \mathit{PassiveVelLimShaping}()$
    **end for**
    $\boldsymbol{w}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_{v_m} \\ -\infty \end{bmatrix},\quad \boldsymbol{w}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_{v_M} \\ \infty \end{bmatrix}$
    $\boldsymbol{H} \leftarrow \begin{bmatrix} \rho \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_m \end{bmatrix}$
    $\boldsymbol{C} \leftarrow \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{I}_m \\ \bar{\boldsymbol{A}} & \boldsymbol{0} \end{bmatrix}$
    $\boldsymbol{b}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_m \\ \bar{\boldsymbol{b}}_m \end{bmatrix},\quad \boldsymbol{b}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_M \\ \bar{\boldsymbol{b}}_M \end{bmatrix}$
    $\boldsymbol{w}^* \leftarrow \begin{bmatrix} \dot{\boldsymbol{q}}_v^* \\ \boldsymbol{s}^* \end{bmatrix} \leftarrow \mathit{solveQP}()$
    $\bar{\boldsymbol{b}}_m \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_m + \boldsymbol{s}^* \\ \bar{\boldsymbol{b}}_m \end{bmatrix},\quad \bar{\boldsymbol{b}}_M \leftarrow \begin{bmatrix} \dot{\boldsymbol{\alpha}}_M + \boldsymbol{s}^* \\ \bar{\boldsymbol{b}}_M \end{bmatrix},\quad \bar{\boldsymbol{A}} \leftarrow \begin{bmatrix} \boldsymbol{A} \\ \bar{\boldsymbol{A}} \end{bmatrix}$
**end for**
**for** $j \leftarrow 0$ *to* $n$ **do**
    $E_{j_r} \leftarrow E_{j_r} + \Delta E_{\text{shuffle}} + E_{j_{D_{\min}}} - \frac{1}{2}K_j[(\Delta q_j + \Delta T \dot{q}_{j_v})^2 - \Delta q_j^2]$
    **if** $E_{j_r} > E_{r_{\max}}$ **then**
        $E_{j_r} \leftarrow E_{r_{\max}}$
    **end if**
**end for**
**return** $\dot{\boldsymbol{q}}_v^*$
```

Bibliography

[1] JJ Abbott and AM Okamura. Effects of position quantization and sampling rate on virtual-wall passivity. *IEEE Transactions on Robotics*, 21(5):952–964, 2005.

[2] ROBERTJ Anderson and Mark W Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal of Robotics and Automation*, 4(5):549–556, 1988.

[3] G. Antonelli. Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems. *IEEE Transactions on Robotics*, 25(5):985–994, October 2009.

[4] Gianluca Antonelli. Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, number 3, pages 5892–5897, 2009.

[5] P Baerlocher and R Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *IEEE/RSJ International Conference on Intelligent Robots and System*, number 2, 1998.

[6] Paolo Baerlocher and Ronan Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6):402–417, June 2004.

[7] J Baillieul. Kinematic programming alternatives for redundant manipulators. In *IEEE International Conference on Robotics and Automation*, pages 722–728, 1985.

[8] LF Baptista, JMC Sousa, and JMGS da Costa. Predictive Force Control of Robot Manipulators in Nonrigid Environments. In *Industrial Robotics: Theory, Modelling and Control*, number December, pages 841–875. 2006.

[9] FA Behringer. Lexicographic quasiconcave multiobjective programming. *Zeitschrift für Operations Research*, pages 103–116, 1977.

[10] Adi Ben-Israel and Thomas NE Greville. *Generalized Inverses*, volume 13. Springer,

2003.

[11] Rainer Bischoff and Johannes Kurth. The KUKA-DLR Lightweight Robot arm-a new reference platform for robotics research and manufacturing. *Robotics (ISR)*, pages 741–748, 2010.

[12] Ronan Boulic, R Mas, and Daniel Thalmann. A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics*, 20(5):693–701, 1996.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[14] Oliver Brock and Oussama Khatib. Elastic strips: A framework for motion generation in human environments. *The International Journal of Robotics Research*, 21(12):1031, 2002.

[15] H. Bruyninckx and J. De Schutter. Specification of force-controlled actions in the "task frame formalism"-a synthesis. *IEEE Transactions on Robotics and Automation*, 12(4):581–589, 1996.

[16] J. Cervera, a.J. van der Schaft, and a. Baños. Interconnection of port-Hamiltonian systems and composition of Dirac structures. *Automatica*, 43(2):212–225, February 2007.

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

[18] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano. Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy. *The International Journal of Robotics Research*, 10(4):410–425, August 1991.

[19] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, June 1997.

[20] Stefan Chiaverini and Lorenzo Sciavicco. The parallel approach to force/position control of robotic manipulators. *Robotics and Automation, IEEE . . .*, 9(4), 1993.

[21] Woojin Chung, Changju Rhee, Youngbo Shim, Hyungjin Lee, and Shinsuk Park. Door-Opening Control of a Service Robot Using the Multifingered Robot Hand. *IEEE Transactions on Industrial Electronics*, 56(10):3975–3984, October 2009.

[22] Richard Colbaugh, Homayoun Seraji, and Kristin Glass. Direct adaptive impedance control of robot manipulators. *Journal of Robotic Systems*, 10(2):217–248, 1993.

[23] TL De Fazio, DS Seltzer, and DE Whitney. The instrumented remote centre compliance. *Industrial robot*, 11(4):238–242, 1984.

[24] J. de Schutter. Estimating First-Order Geometric Parameters and Monitoring Contact Transitions during Force-Controlled Compliant Motion. *The International Journal of Robotics Research*, 18(12):1161–1184, December 1999.

[25] J. De Schutter, T. De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, E. Aertbeliën, Kasper Claes, and H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433, 2007.

[26] J. De Schutter and H. Van Brussel. Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks. *The International Journal of Robotics Research*, 7(4):3–17, August 1988.

[27] J. De Schutter and H. Van Brussel. Compliant Robot Motion II. A Control Approach Based on External Control Loops. *The International Journal of Robotics Research*, 7(4):18–33, August 1988.

[28] W. Decré, Ruben Smits, Herman Bruyninckx, and J. De Schutter. Extending iTaSC to support inequality constraints and non-instantaneous task specification. In *International Conference on Robotics and Automation, 2009. ICRA'09. IEEE*, pages 964–971. IEEE, 2009.

[29] Arati S. Deo and Ian D. Walker. Overview of damped least-squares methods for inverse kinematics of robot manipulators. *Journal of Intelligent & Robotic Systems*, 14(1):43–68, September 1995.

[30] Rosen Diankov, S.S. Srinivasa, Dave Ferguson, and James Kuffner. Manipulation planning with caging grasps. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 285–292. IEEE, 2008.

[31] Alexander Dietrich and T Wimbock. Integration of reactive, torque-based self-

collision avoidance into a task hierarchy. *IEEE Transactions on Robotics*, 2012.

[32] Joseph Duffy. The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces. *Journal of Robotic Systems*, 7(2):139–144, 1990.

[33] O Egeland. Task-space tracking with redundant manipulators. *Robotics and Automation, IEEE Journal of*, (5):471–475, 1987.

[34] a. Escande, N. Mansard, and P.-B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, May 2014.

[35] Adrien Escande, Nicolas Mansard, and P.B. Wieber. Fast Resolution of Hierarchized Inverse Kinematics with Inequality Constraints. In *IEEE International Conference on Robotics and Automation*, number 4, pages 3733–3738, 2010.

[36] Bernard Faverjon and Pierre Tournassoud. A local based approach for path planning of manipulators with a high number of degrees of freedom. In *IEEE International Conference on Robotics and Automation*, pages 1152–1159, 1987.

[37] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A Tank-Based Approach to Impedance Control with Variable Stiffness. In *IEEE International Conference on Robotics and Automation*, pages 4933–4938, 2013.

[38] HJ Ferreau. qpOASES user's manual. Technical Report March, 2011.

[39] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[40] Fabrizio Flacco and Alessandro De Luca. A reverse priority approach to multi-task control of redundant robots. In *IEEE/RSJ International Conference on Intelligent Robots and System*, 2014.

[41] Fabrizio Flacco, Alessandro De Luca, and O Khatib. Motion control of redundant robots under joint constraints: Saturation in the null space. *IEEE International Conference on Robotics and Automation*, 2012.

[42] Matteo Fumagalli, Serena Ivaldi, Marco Randazzo, Lorenzo Natale, Giorgio Metta, Giulio Sandini, and Francesco Nori. Force feedback exploiting tactile and proximal

force/torque sensing. *Autonomous Robots*, 33(4):381–398, April 2012.

[43] Hideo Hanafusa, Tsuneo Yoshikawa, and Yoshihiko Nakamura. Analysis and control of articulated robot with redundancy. In *IFAC, 8th Triennal World Congress*, volume 4, pages 1927–1932, 1981.

[44] B. Hannaford. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002.

[45] N. Hogan. On the stability of manipulators performing contact tasks. *IEEE Journal on Robotics and Automation*, 4(6):677–686, 1988.

[46] Neville Hogan. Impedance control: An approach to manipulation: Part iâĂŤiii. *Journal of dynamic systems, measurement, and control*, 107(1):1–24, 1985.

[47] Advait Jain and C.C. Kemp. Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control. In *IEEE International Conference on Robotics and Automation*, pages 1807–1814. IEEE, 2010.

[48] Advait Jain and Charles C. Kemp. Pulling open novel doors and drawers with equilibrium point control. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 498–505. Ieee, December 2009.

[49] S. Jung. Force Tracking Impedance Control for Robot Manipulators with an Unknown Environment: Theory, Simulation, and Experiment. *The International Journal of Robotics Research*, 20(9):765–774, September 2001.

[50] S. Jung, T.C. Hsia, and R.G. Bonitz. Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment. *IEEE Transactions on Control Systems Technology*, 12(3):474–483, May 2004.

[51] Oussama Kanoun, Florent Lamiraux, and Pierre-Brice Wieber. Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task. *IEEE Transactions on Robotics*, 27(4):785–792, August 2011.

[52] Yiannis Karayiannidis and Christian Smith. “Open Sesame!” Adaptive Force/Velocity Control for Opening Unknown Doors. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 4040–4047, 2012.

[53] Yiannis Karayiannidis, Christian Smith, FE Vina, Petter Ogren, and Danica

Kragic. Model-free robot manipulation of doors and drawers by means of fixed-grasps. In *IEEE International Conference on Robotics and Automation*, pages 4470–4477, 2013.

[54] Homayoon Kazerooni, Paul K Houpt, and Thomas B Sheridan. Robust compliant motion for manipulators, part i: the fundamental concepts of compliant motion. *IEEE Journal of Robotics and Automation*, 2(2):83–92, 1986.

[55] O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, pages 90–98, 1986.

[56] O Khatib. The potential field approach and operational space formulation in robot control. *Adaptive and Learning Systems*, 1986.

[57] O Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 1987.

[58] JP Kim and J Ryu. Stable haptic interaction control using energy bounding algorithm. *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 1210–1217, 2004.

[59] T.a. Lasky and T.C. Hsia. On force-tracking impedance control of robot manipulators. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, (April):274–280, 1991.

[60] Dongjun Lee and Ke Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Transactions on Robotics*, 26(2):354–369, 2010.

[61] Jaemin Lee, Nicolas Mansard, and Jaeheung Park. Intermediate desired value approach for continuous transition among multiple tasks of robots. *IEEE International Conference on Robotics and Automation*, (3):1276–1282, 2011.

[62] T. Lefebvre, H. Bruyninckx, and J. De Schutter. Polyhedral contact formation modeling and identification for autonomous compliant motion. *IEEE Transactions on Robotics and Automation*, 19(1):26–41, February 2003.

[63] Tine Lefebvre, Jing Xiao, Herman Bruyninckx, and Gudrun de Gersem. Active compliant motion: a survey. *Advanced Robotics*, 19(5):479–499, June 2005.

[64] A Liegeois. Automatic supervisory control of the configuration and behavior of

multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977.

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

[66] Ziren Lu and Andrew A Goldenberg. Robust impedance control and force regulation: theory and experiments. *The International journal of robotics research*, 14(3):225–254, 1995.

[67] Ewald Lutscher and Gordon Cheng. A set-point-generator for indirect-force-controlled manipulators operating unknown constrained mechanisms. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4072–4077. Ieee, October 2012.

[68] Ewald Lutscher, Martin Lawitzky, Gordon Cheng, and Sandra Hirche. A control strategy for operating unknown constrained mechanisms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 819–824. IEEE, 2010.

[69] Dedi Ma, Hesheng Wang, and Weidong Chen. Unknown constrained mechanisms operation based on dynamic hybrid compliance control. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 2366–2371. Ieee, December 2011.

[70] a. a. Maciejewski and C. a. Klein. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments. *The International Journal of Robotics Research*, 4(3):109–117, September 1985.

[71] AA Maciejewski. Dealing with the ill-conditioned equations of motion for articulated figures. *Computer Graphics and Applications, IEEE*, pages 63–71, 1990.

[72] N. Mansard, O. Khatib, and A. Kheddar. A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks. *IEEE Transactions on Robotics*, 25(3):670–685, June 2009.

[73] Nicolas Mansard and Olivier Stasse. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *International Conference on Advanced Robotics, 2009. ICAR*, number 8, 2009.

[74] Matthew T. Mason. Compliance and Force Control for Computer Controlled Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432,

1981.

[75] Wim Meeussen, Melonee Wise, Stuart Glaser, Sachin Chitta, C. McGann, Patrick Mihelich, E. Marder-Eppstein, Marius Muja, Victor Eruhimov, Tully Foote, and Others. Autonomous door opening and plugging in with a personal robot. In *International Conference on Robotics and Automation (ICRA), 2010 IEEE*, pages 729–736. IEEE, 2010.

[76] R.M. Murray, Z. Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994.

[77] Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement, and control*, 108(3):163–171, 1986.

[78] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3, 1987.

[79] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3901–3908, 2005.

[80] BJ Nelson and PK Khosla. Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. *The International journal of robotics ...*, pages 255–269, 1995.

[81] Dragomir N Nenchev. Redundancy resolution through local optimization: A review. *Journal of Robotic systems*, 6(6):769–798, 1989.

[82] J.L. Nevins and D.E Whitney. *The Force Vector Assembler Concept*. Springer, 1972.

[83] Gunter Niemeyer and J.J.E. Slotine. A simple strategy for opening an unknown door. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 2, pages 1448–1453. IEEE, 1997.

[84] M Peinado, R Boulic, B Le Callennec, and D Méziat. Progressive cartesian inequality constraints for the inverse kinematic control of articulated chains. *Proc. of Eurographics*, pages 2–5, 2005.

[85] Michael A Peshkin. Programmed compliance for error corrective assembly. *IEEE Transactions on Robotics and Automation*, 6(4):473–482, 1990.

[86] L. Peterson, D. Austin, D. Kragic, and Lars Petersson. High-level control of a mobile manipulator for door opening. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, pages 2333–2338. Ieee, 2000.

[87] Anna Petrovskaya and A.Y. Ng. Probabilistic Mobile Manipulation in Dynamic Environments, with Application to Opening Doors. In *IJCAI*, pages 2178–2184. Morgan Kaufmann Publishers Inc., 2007.

[88] Mario Prats, Steven Wieland, Tamim Asfour, A. del Pobil, and R. Dillmann. Compliant interaction in household environments by the Armar-III humanoid robot. In *8th IEEE-RAS International Conference on Humanoid Robots*, pages 475–480, 2008.

[89] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *IEEE International Conference on Robotics and Automation*, pages 802–802. Citeseer, 1993.

[90] MH Raibert and JJ Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, pages 126–133, 1981.

[91] Daniel Raunhardt and Ronan Boulic. Progressive clamping. In *IEEE International Conference on Robotics and Automation*, number April, pages 10–14, 2007.

[92] Changju Rhee, Munsang Kim, and Hyungjin Lee. Door opening control using the multi-fingered robotic hand for the indoor service robot. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 4011–4016 Vol.4. Ieee, 2004.

[93] T Ruhr, J Sturm, and D Pangercic. A generalized framework for opening doors and drawers in kitchen environments. In *IEEE International Conference on Robotics and Automation*, pages 3852–3858, 2012.

[94] Jee Hwan Ryu, Dong Soo Kwon, and Blake Hannaford. Stable teleoperation with time-domain passivity control. *IEEE Transactions on Robotics and Automation*, 20(2):365–373, 2004.

[95] JH Ryu, YS Kim, and Blake Hannaford. Sampled- and continuous-time passivity and stability of virtual environments. *IEEE Transactions on Robotics*, 20(4):772–

776, 2004.

[96] JK Salisbury. Active stiffness control of a manipulator in Cartesian coordinates. In *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980.

[97] Giulio Sandini, Giorgio Metta, and David Vernon. The icub cognitive humanoid robot: An open-system research platform for enactive cognition. In *Neural Networks*, pages 1125–1134. 2010.

[98] Arjan van der Schaft and AJ Schaft. *L2-gain and passivity in nonlinear control.* Springer-Verlag New York, Inc., 1999.

[99] Joseph M Schimmels and Michael A Peshkin. Admittance matrix design for force-guided assembly. *IEEE Transactions on Robotics and Automation*, 8(2):213–227, 1992.

[100] A.J. Schmid, Nicolas Gorges, D. Goger, and H. Worn. Opening a Door with a Humanoid Robot Using Multi-Sensory Tactile Feedback. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 285–291. IEEE, 2008.

[101] Luis Sentis and Oussama Khatib. Task-oriented control of humanoid robots through prioritization. In *IEEERAS/RSJ International Conference on Humanoid Robots*, pages 1–16, 2004.

[102] Luis Sentis and Oussama Khatib. Control of free-floating humanoid robots through task prioritization. *IEEE International Conference on Robotics and Automation*, (April):18–23, 2005.

[103] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):1–15, 2005.

[104] H. Seraji and R. Colbaugh. Force Tracking in Impedance Control. *The International Journal of Robotics Research*, 16(1):97–117, February 1997.

[105] B. Siciliano and J.-J.E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pages 1211–1216, 1991.

[106] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics.* Springer

Science & Business Media, 2008.

[107] Bruno Siciliano and Luigi Villani. *Robot force control*. Springer, 1999.

[108] S.K. Singh and D.O. Popa. An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments. *IEEE Transactions on Robotics and Automation*, 11(6):912–921, 1995.

[109] B. Stanczyk. *Development and Control of an Anthropomorphic Telerobotic System*. PhD thesis, Technische Universität München, 2006.

[110] J Sturm, A Jain, and Cyrill Stachniss. Operating articulated objects based on experience. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 2739–2744, 2010.

[111] YW Sung, DK Cho, and MJ Chung. A constrained optimization approach to resolving manipulator redundancy. *Journal of robotic systems*, 1996.

[112] Patrizio Tomei. A simple PD controller for robots with elastic joints. *Automatic Control, IEEE Transactions on*, 36(10):1208–1213, 1991.

[113] Luigi Villani, Carlos Canudas de Wit, and Bernard Brogliato. An exponentially stable adaptive control for force and position tracking of robot manipulators. *IEEE Transactions on Automatic Control*, 44(4):798–802, 1999.

[114] CW Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, (1):93–101, 1986.

[115] Daniel E Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, 99(2):91–97, 1977.